



UNIVERSITAT DE
BARCELONA

Trabajo final de grado

GRADO DE INGENIERÍA
INFORMÁTICA

Facultad de Matemáticas e Informática
Universidad de Barcelona

WorldEditor para Fracsland

Autor: Luna García Villa

Director: Dra. Anna Puig Puig
Realizado en: Departamento de
Matemáticas e Informática. UB

Barcelona, 21 de junio de 2017

Abstract

WorldEditor for Fracsland is an application that allows to extend the game of Fracsland. Fracsland is a *Serious Game* in development that allows you to learn the concept of fractions in a fun and enjoyable way using a game.

Serious Games are games developed mainly to teach a concept or idea, through a video game to make it more fun and enjoyable. In this case, it consists of the player learning mathematical concepts focused on upper elementary school students.

Most games usually have a reduced game life, since once important events of the game are completed, the player usually stops playing because he tires to continue playing.

The application developed in this project has been developed using the graphic engine *Unity 5*, which allows to extend the game of Fracsland without needing to have technical knowledge, through the creation of new scenarios and challenges, allowing to play with this scenario In the original game.

Resum

WorldEditor per Fracsland és una aplicació que permet ampliar el joc de Fracsland. Fracsland és un joc en desenvolupament de tipus *Serious Game* el que permet aprendre el concepte de fraccions de forma divertida i amena utilitzant un joc.

Els Serious Games són jocs desenvolupats principalment per ensenyar un concepte o idea, a través d'un videojoc perquè sigui més amè i divertit. En aquest cas, consisteix que el jugador aprengui conceptes matemàtics enfocat a alumnes de cicle superior de primària.

La majoria de jocs solen tenir una vida de joc reduïda, ja que una vegada s'ha completat els esdeveniments importants del joc, el jugador sol deixar de jugar perquè es cansa per seguir jugant.

L'aplicació desenvolupada en aquest projecte s'ha desenvolupat usant el motor gràfic *Unity 5*, que permet ampliar el joc de Fracsland sense la necessitat de tenir coneixements tècnics, a través de la creació de nous escenaris i reptes, permetent jugar amb aquest escenari en el joc original.

Resumen

WorldEditor para Fracsland es una aplicación que permite ampliar el juego de Fracsland. Fracsland es un juego en desarrollo de tipo *Serious Game* lo que permite aprender el concepto de fracciones de forma divertida y amena usando un juego.

Los Serious Games son juegos desarrollados principalmente para enseñar un concepto o idea, a través de un videojuego para que sea más ameno y divertido. En este caso, consiste en que el jugador aprenda conceptos matemáticos enfocado a

alumnos de ciclo superior de primaria.

La mayoría de juegos suelen tener una vida de juego reducida, ya que una vez se ha completado los eventos importantes del juego, el jugador suele dejar de jugar porque se cansa para seguir jugando.

La aplicación desarrollada en este proyecto se ha desarrollado usando el motor gráfico *Unity 5*, que permite ampliar el juego de Fracslan sin necesidad de tener conocimientos técnicos, a través de la creación de nuevos escenarios y retos, permitiendo jugar con este escenario en el juego original.

Agradecimientos

Quiero agradecer a mi tutora Anna Puig por su apoyo y su ayuda a lo largo del proyecto.

A toda mi familia por la comprensión que me han dado, en especial a mis padres y hermanos.

Y a todos mis amigos por el apoyo y ánimos que me han ido dando.

Índice

1. Introducción	1
1.1. Motivación	1
1.2. Juegos serios y fracciones	2
1.3. Objetivo general de proyecto	2
1.4. Objetivos específicos del proyecto	3
1.5. Planificación	4
1.6. Organización de la memoria	7
2. Antecedentes	8
2.1. Juegos matemáticos	8
2.1.1. Juegos tradicionales	8
2.1.2. Juegos matemáticos digitales	9
2.1.3. Videojuegos de fracciones	9
2.2. Mods en juegos matemáticos o similares	11
2.3. Conclusión	12
3. Análisis	13
3.1. Fracsland	13
3.1.1. Introducción	13
3.1.2. Fracciones	13
3.1.3. Zonas de Fracsland	14
3.1.4. Misiones o retos	16
3.1.5. Componentes de las misiones o retos	17
3.2. ¿Qué mejorar?	18
3.2.1. Mapa	18
3.2.2. Misiones	19
3.3. Elementos editables y no editables	20
3.3.1. Elementos editables	20
3.3.1.1. Elementos obligatorios (Work Objects)	20
3.3.1.2. Elementos opcionales	21
3.3.2. Elementos no editables	21
3.4. ¿Cómo mejorar el juego?	21
3.4.1. Menú principal	22

3.4.2.	Menú de <i>Work Objects</i>	24
3.4.3.	Menú lateral	25
4.	Diseño	26
4.1.	Draft de la interficie y su funcionamiento	26
4.1.1.	Flujo de la UI	26
4.1.2.	Casos de uso	26
4.1.2.1.	Primer grupo (Color verde)	27
4.1.2.2.	Segundo grupo (Color azul)	28
4.1.2.3.	Tercer grupo (Color morado)	29
4.1.2.4.	Cuarto grupo (Color rojo)	30
4.2.	Diagrama de dominio	30
4.3.	Insertar elementos en el escenario	33
4.3.1.	Funcionamiento general	33
4.3.2.	Estrategia utilizada	34
4.4.	Herramientas utilizadas	35
4.4.1.	Work Objects	36
4.4.2.	Elementos decorativos y NPCs	36
4.4.3.	Texturas y colores	37
4.4.4.	Guardar y cargar una escena	37
5.	Implementación y simulaciones	38
5.1.	Implementación	38
5.1.1.	Diagrama de clases	38
5.2.	Funcionamiento del juego	40
5.2.1.	Introducción al juego	40
5.2.2.	Añadir elementos a la escena	40
5.2.3.	Probar la escena	42
5.2.4.	Guardar y cargar escena	43
5.3.	Resultados de los test de usuario	43
5.3.1.	Resultados del mod	43
5.3.1.1.	Preguntas de respuesta numérica	44
5.3.1.2.	Resultados de las preguntas de redactar	46
6.	Conclusiones y trabajo futuro	47

7. Apéndice A: Manual Técnico	49
7.1. Instalación del entorno de desarrollo: Requisitos mínimos y pasos a seguir	49
7.1.1. Requisitos mínimos	49
7.1.2. Instalación entorno de desarrollo del juego	50
7.1.3. Importar el juego a la versión de Unity	51
7.2. Instalación del juego para usuarios	51

1. Introducción

1.1. Motivación

Los *Serious Games* son juegos enfocados a ayudar en la comprensión de conceptos o materias[2]. Existen diferentes tipos de *Serious Games*, el más popular es el de uso educativo[13]; también existen con fines comerciales, concienciación[11], denuncia social, etcétera.

La idea principal de un *Serious Game* es la utilización de un videojuego para un fin diferente del entretenimiento. En el concepto educativo, se ha utilizado para motivar el aprendizaje de conceptos o ideas que pueden resultar aburridos o tediosos de realización y/o comprensión.

A ciertas personas les cuesta asimilar algunos conceptos, lo que causa que pasado un tiempo, los olviden. Los juegos educativos facilitan técnicas de nemotecnia para aprender estos conceptos, sin tener que memorizar, ayudando a entender el concepto y experimentando con el mismo en un entorno explicado. Todo esto se hace con un toque de diversión, de forma que es más atractivo el proceso de aprendizaje para el usuario final.

Así, aunque existen juegos serios que son muy útiles en el proceso de aprendizaje, al no ser configurables ni extensibles, tienen una vida corta de juego y de aplicación en el aula. Una forma de alargar esta vida es la posibilidad de crear elementos nuevos que modifican algunos aspectos originales del juego original. A esta posibilidad de crear nuevas funcionalidades se le llama *mod* [14]. No todos los juegos tienen esta capacidad, y si la tienen, suele ser limitada.

En el mundo del videojuego educativo existen distintos tipos de conceptos a enseñar, ya sea por la edad para la que se enfoca el juego o bien el tema del cual tratan. Existe el juego que pretende reforzar las lenguas o aprender una nueva [8]. Puede enseñar a niñas y niños pequeños a programar y de esta forma introducirlos en el mundo de la informática[26][7]. Otro tipo de juegos es el que está enfocado a las matemáticas[3].

El proyecto actual surge de la idea y necesidad de extender juegos serios que estén orientados al aprendizaje de las matemáticas de forma fácil. Este aprendizaje ha de ser fácil para el alumno como para el profesor y sin la necesidad de tener conocimientos técnicos de programación.

1.2. Juegos serios y fracciones

En el ámbito de las matemáticas hay juegos que pretenden que la persona aprenda operaciones aritméticas básicas (suma, resta, multiplicación o división)[19]. Otros están enfocados a un público de mayor edad, como el refuerzo de resolver ecuaciones[1].

Este proyecto se ha basado en un juego en desarrollo enfocado en el aprendizaje de fracciones matemáticas a través de misiones. El juego base se llama Fracsland desarrollado por Cristian Muriel Ordóñez[21] como proyecto Final de Grado en el curso 2015-16. En este juego se puede aprender el concepto de fracción mediante la superación de retos integrados en el juego. Al mismo tiempo que el jugador puede explorar un mundo abierto o enfrentarse a monstruos, entre otras aventuras.



Figura 1: Fracsland: Bridge[21]

1.3. Objetivo general de proyecto

En este proyecto se pretende alargar la vida útil del juego, dando al jugador la posibilidad de ampliar el juego. Esta modificación será hecha por usuarios no programadores, es decir, por los propios jugadores. Los jugadores son niños y niñas entre 10 y 12 años. La ampliación de juego permitirá crear nuevos retos a partir de elementos gráficos disponibles. Estos elementos gráficos pueden ser modelos 3D, animaciones, texturas, efectos especiales, etcétera.

En resumen, la ampliación del juego debe permitir diferentes tipos de ediciones:

- Seleccionar e insertar un mapa ya existente, de entre un conjunto de mapas, a la escena.
- Seleccionar y añadir un posible reto matemático que se deberá completar.

Los retos son misiones que podrá completar un usuario, dichos retos están formados por diferentes partes. En Fracsland los retos son problemas matemáticos que para ser solucionados se debe decidir que fracción se ha de usar para completar un todo.

Por ejemplo: un reto incluido en el juego, consiste en terminar de construir un puente mediante tablones. Para conseguir los tablones el jugador debe talar árboles. Una vez tiene los tablones, puede terminar el puente insertando la porción de madera que falta. La porción de madera se selecciona a partir de una interficie en la que el jugador dispone de distintas porciones, de tipo numérico o de fracciones. Esta última parte se considera como una parte lógica para el desarrollo del reto.

Además los retos tienen asociados efectos especiales y/o visuales. Estos efectos se activan al completar un reto correctamente o al fracasar, dependiendo del caso será un efecto u otro.

Estos retos, considerando como retos a la unión de las partes necesarias para llevar a cabo el reto, la lógica y los efectos, a seleccionar ya están creados.

- Añadir recursos adicionales (o ya existentes en el juego original) que permitan completar el reto, o elementos de ámbito decorativo.
- Insertar a la escena personajes no jugadores (*NPCs*): pueden ser humanoides y/o animales con los que interactuar.
- Insertar al reto uno o varios efectos especiales disponibles en un conjunto de FX prediseñados.
- Dar la posibilidad al jugador de modificar ciertas texturas de algunos elementos añadidos a la escena .

De esta forma se pretende que el juego pueda llegar a un público mayor y que no se convierta en un juego cerrado y con una vida corta, sino dinámico y abierto, con muchas más posibilidades pedagógicas.

1.4. Objetivos específicos del proyecto

El objetivo general se puede desglosar en los siguientes objetivos específicos.

- Estudio del motor gráfico *Unity 5*[29] y la arquitectura inicial de Fracslan: Cómo funciona internamente Unity y cómo se ha de trabajar con él. Por el lado de Fracslan, entender como se desarrolló y qué elementos lo componen.
- Análisis de los componentes del juego que puedan ser modificables por el jugador, así como aquellos elementos que se ofrecerán ya creados porque representan un todo. Por ejemplo: como se ha mencionado anteriormente, los retos conllevan una serie de partes, algunas de esas partes son inseparables: la parte física del reto y la lógica. Esto puede ser un elemento ya creado, pero pueden haber más objetos.

De esta forma es posible tener una idea de la cantidad de elementos que podrá tener el jugador a su disposición a la hora de crear la escena.

- Diseñar una interfaz amigable, intuitiva y de fácil uso para niños de 10-12 años. Con esta interfaz se ha de poder insertar y trabajar con todos los elementos modificables de Fracslan.
- Análisis, diseño e implementación de un sistema que permita la edición de escenarios. Todos los componentes analizados, que son modificables, se han de poder añadir a la escena.
- Desarrollo de la aplicación aplicando todo lo analizado y diseñado. Y crear un entorno que permita la integración de Fracslan en este proyecto.

1.5. Planificación

Para desarrollar los objetivos específicos definidos en el apartado anterior se ha optado en crear el desarrollo basado en un diseño ágil. El diseño ágil que se usó es SCRUM, de esta manera el proyecto se ha dividido en historias de usuario.

1. Aprendizaje Unity 5 a través de tres tutoriales: *Tutorial Space Shooter*[30], *Tutorial Adventure Game*[29] y *Tutorial UI*[28].
2. Familiarización con Fracslan[21].
3. Crear una UI de pruebas para estudiar cómo modificar el cielo de la escena.
4. Diseñar una UI para el mod que sea amigable.
5. Añadir al mod aquellos elementos que queremos que sean modificables o necesarios para completar el mapa, por ejemplo: insertar un reto o añadir árboles en caso de necesitar madera a la hora de completar el reto.

6. Crear una *grid*[17] para poder insertar los objetos en el mapa y *Drag & Drop* de los elementos a insertar
7. Insertar en el mod elementos de Fracsland para integrar ambos juegos.
8. Ejecutar el mapa que se ha creado y volver al editor del mapa.
9. Guardar y cargar la escena en un JSON.

Para llevar a cabo las historias de usuario, se ha diseñado una planificación (ver la *fig: 2*) que permite crear distintas tareas a partir las historias. De las cuales algunas son cíclicas y otras son secuenciales poder desarrollarlas. Para el desarrollo de esta planificación se ha estimado un tiempo de prácticamente 4 meses, desde Febrero a mediados de Junio.

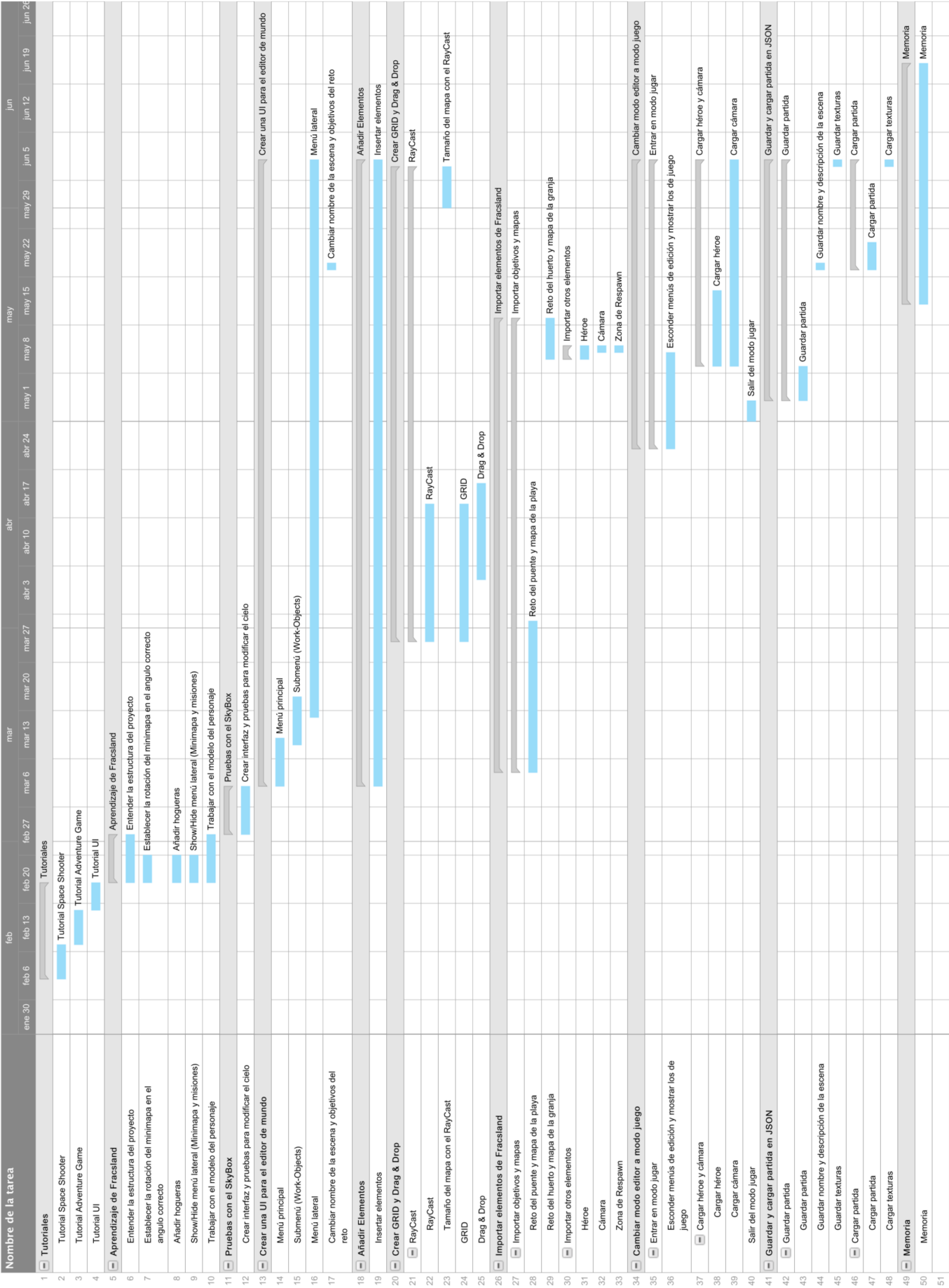


Figura 2: Diagrama de Gantt

1.6. Organización de la memoria

La memoria se ha organizado en los siguientes apartados:

1. **Introducción:** En este capítulo se hace una breve presentación del proyecto diciendo qué es, cuál es su origen, cuáles son sus objetivos y cuáles son las tareas que se realizarán para llevar a cabo estas tareas.
2. **Antecedentes:** En este apartado se explica de forma breve la historia de los juegos matemáticos y la evolución de los mods.
3. **Análisis:** En este apartado se analiza el juego desarrollado en un proyecto anterior, llamado Fracsland. Y que elementos pueden ser mejorados para solucionar el problema que se ha planteado: ¿Cómo alargar la vida del juego?. Además se expone como se ha planteado resolverlo.
4. **Diseño:** En este capítulo se explica cómo se ha diseñado el mod para que cumpla con ciertas condiciones, como: la usabilidad para niños y niñas de 10-12.

Se explica como ha sido diseñada la interfaz gráfica hasta llegar a la estructura necesaria para llevar a cabo este mod.

5. **Implementación y simulaciones:** Aquí se muestra cómo se ha implementado los diseños y planificaciones hechos, a partir de una simulación.
Y asimismo, los resultados de las simulaciones hechas, además se incluyen los resultados de los test de usuario realizados.
6. **Conclusiones y trabajo futuro:** En este apartado se explica cuál era el objetivo principal, así como lo que se ha conseguido y futuras ampliaciones.
7. **Apéndice:** En este capítulo se detallan los requisitos técnicos, se explica qué y cómo se ha de instalar para desarrollar este mod. También qué ha de hacer un usuario final para instalar el juego y cómo jugar.

2. Antecedentes

Para entender el significado de crear un videojuego con ámbito matemático, se debe entender la evolución que han tendido los juegos matemáticos a lo largo de la historia.

De la misma manera es importante entender que quiere decir crear una modificación de un juego existente y cual han sido su evolución.

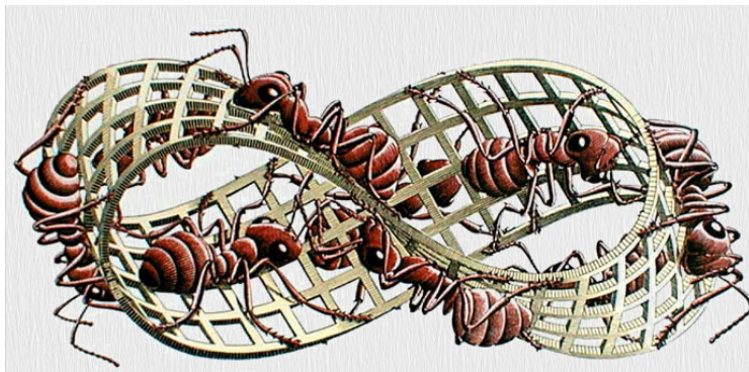
2.1. Juegos matemáticos

En primer lugar se va explicar la evolución de los juegos relacionados con las matemáticas, desde los más tradicionales a los actuales videojuegos disponibles en la red.

2.1.1. Juegos tradicionales

De juegos matemáticos existen muchos tipos, formas de transmitirlos o formas de realizar el juego. Existen juegos más clásicos que suelen venir en libros, ya que normalmente hay que resolver problemas y/o jugar con objetos tipo *Tangram*[24]. El Tangram es un juego en el que a partir de triángulos y cuadriláteros se deben realizar diferentes figuras geométricas.

Un juego que se suele enseñar, pero que también se ha usado en aplicaciones reales, son las Bandas de Moebius[9](fig:3). La Banda de Moebius es una cinta que sin tener que levantar el dedo de ella, la persona pasa del lado exterior de la banda al interior y viceversa, en otras palabras, solo tienen una cara. Este descubrimiento lo hizo el matemático alemán *August Ferdinand Möbius*.



(a) Banda de Moebius

5	3			7			
6			1	9	5		
	9	8				6	
8				6			3
4			8		3		1
7				2			6
	6					2	8
			4	1	9		5
				8			7
						7	9

(b) Ejemplo de Sudoku

Figura 3: Banda de Moebius[25] y Ejemplo de Sudoku[32]

Un juego muy famoso de ámbito matemático es el *Sudoku*[32](fig:3). Este juego consiste en rellenar una matriz 9x9 con números del 1 al 9, pero los números no se pueden repetir ni en filas, columnas o en sub-cuadrados de 3x3.

2.1.2. Juegos matemáticos digitales

Con la aparición de las nuevas tecnologías se ha comenzado a desarrollar otro tipo de juegos matemáticos; los que requieren de un ordenador, *Tablet* o *Smartphone*.

Un ejemplo de juego desarrollado para ordenador y para enseñar matemáticas básicas es *Mathbreakers*[3]. Este juego consiste en resolver operaciones matemáticas haciendo que el usuario se enfrente a obstáculos y enemigos, que están hechos de números. Para derrotar al enemigo o los obstáculos hay que dar el resultado correcto de una operación matemática.



Figura 4: Mathbreakers[19] y Multiplications with Math Mathews [16] [4]

Por otro lado *Multiplications with Math Mathews*[16] es un juego desarrollado para *Tablets* y/o *Smartphones* por *Kiupe*, pensado para aprender las tablas de multiplicar. Este juego tiene tres tipos de situaciones para practicar las tablas de multiplicación. El juego consiste en encontrar la cura para el capitán Mathews que es víctima de una maldición.

2.1.3. Videojuegos de fracciones

En el ámbito de los videojuegos matemáticos existe un pequeño apartado dedicado al aprendizaje de fracciones. A continuación se exponen 4 juegos *flash* que se han encontrado disponibles de forma *on-line* en la red como ejemplos de la integración entre el aprendizaje del concepto de fracciones y un juego. Algunos de estos

juegos visualizan el concepto de fracción a partir de los números, otros, en cambio, lo hacen a partir de símbolos gráficos en forma de pastel, de tabletas, etc.

Pipoclub[23] consiste en ayudar a un aviador a llegar a su destino. Para conseguirlo el juego plantea una operación de fracciones y el jugador debe seleccionar el resultado correcto. Cuando se hace correctamente, el avión continua su vuelo.

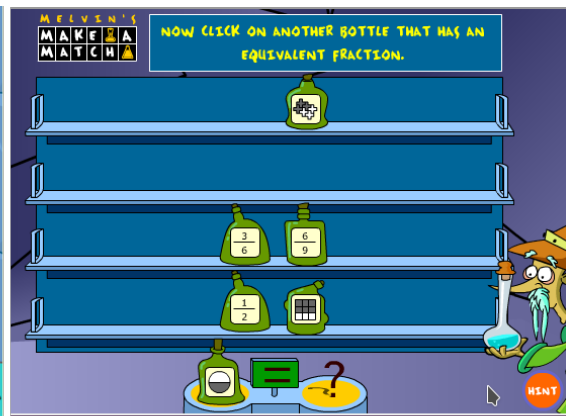
Por otro lado, *Malvin*[22] es un juego que trata de encontrar el valor equivalente de una fracción con el dibujo que lo representa. Por ejemplo: se ve una botella medio llena, la fracción que la representa es $1/2$.

Una forma de aprender qué fracciones tienen más peso que otras, es con el siguiente juego: *Dolphin Racing*[15]. Este juego consiste en seleccionar una fracción de las que se muestran y en función del valor de esta fracción, el delfín avanzará más o menos.

Por último está *Problem Solving*[18] que consiste en crear una especie de puente usando fracciones. En el ejemplo de la figura 5d hay que salvar un hueco de 1m, para poder cruzar se debe escoger qué fracciones rellenan este metro.



(a) Pipoclub Fracciones



(b) Melvin



(c) Dolphin Racing



(d) Problem Solving

Figura 5: Pipoclup[23], Melvin[22], Dolphin Racing[12] y Problem Solving[18]

Por lo que se ha podido estudiar, este tipo de juegos motivan al estudiante a seguir aprendiendo por su forma fácil de jugar y de entender el concepto de las fracciones.

A partir de esta idea surgió el propósito de desarrollar Fracslan, desde el que era posible jugar con fracciones mediante diferentes representaciones gráficas. Estas representaciones estaban apoyadas con una representación numérica las fracciones.

2.2. Mods en juegos matemáticos o similares

La posibilidad de crear modificaciones a videojuegos es bastante antigua. Uno de los primeros juegos que daba esta posibilidad fue el *Warcraft III*[5], desarrollado por *Blizzard Entertainment*[10]. Este juego incluía un editor interno con el que se podía crear un mapa usando los elementos del juego original. También era posible editar un poco de código para que la modificación funcionara mejor.

Un mod famoso de este juego, que posteriormente se convirtió en un juego nuevo y abrió un nuevo tipo de juego, fue el *DOTA*[31] y el tipo de juego que creó fue el *MOBA*[27].

Tras el éxito del *Warcraft III* con las modificaciones, llegaron otros juegos que además de crear nuevos mapas, daba la posibilidad de cambiar las texturas del juego, crear una nueva interfaz, añadir ayudas, etc.

Un juego famoso por ser editable es *Word of Warcraft*[6], también de *Blizzard Entertainment*. Permitía modificar la interfaz de usuario, insertar ayudas, etc.

Por otro lado, otro juego famoso por su alto nivel de modificación es *Minecraft*[20], creado por *Mojang*. Este juego permitía crear mapas, cambiar las texturas del juego, modificar el *Skin* del personaje, etc.

En los *Serious Games* han habido ampliaciones de los juegos, pero principalmente por el lado del *authoring* principalmente enfocado a los profesores. Estos sistemas de *authoring* permiten configurar los juegos, así como añadir preguntas y respuestas, pero en general permiten poca edición del juego en sí.

En el proyecto de Fracslan se ofrecía al profesor la posibilidad de configurar las misiones modificando las fracciones a aprender o la que representación gráfica usar, pero no era posible configurar las escenas, crear nuevos retos o nuevos escenarios.

2.3. Conclusión

Observando que muchos de los juegos más exitosos tienen la opción de crear mods, pero en cambio existen pocos juegos serios que den la oportunidad de editar el propio juego por los propios estudiantes, sino lo que suelen permitir es modificar el authoring. Por esta razón se ha considerado crear la opción de desarrollar un mod para Fracslan. Con este mod se pretende dar la posibilidad de crear nuevos escenarios o retos nuevos. Lo cual, de esta manera se espera poder alargar la vida útil del juego.

3. Análisis

3.1. Fracsland

3.1.1. Introducción

Como se ha mencionado anteriormente, este proyecto ofrece un mod para poder ampliar el juego de Fracsland[21].

Fracsland es un juego de aventuras para un solo jugador. El mundo de Fracsland permite recorrer el mapa libremente, es decir es un juego de mundo abierto, en este mundo existen diferentes retos y dificultades que el jugador podrá ir descubriendo.

El jugador aparece en una isla desierta y su objetivo es escapar de esta isla. Para escapar es necesario conseguir todas las partes de un motor para una balsa. El jugador irá consiguiendo las distintas piezas del motor a medida que consiga materiales y vaya pasando exitosamente los retos que se irán planteando durante el juego.

En la isla hay un pequeño pueblo, en este pueblo hay una persona que se llama *Lord Barus*. Lord Barus irá dando pequeñas misiones al superviviente, en las que deberá ayudar a la pobre gente de la isla. A medida que vaya completando misiones obtendrá piezas para poder completar el motor de la balsa, de la misma forma estará aprendiendo nuevas fracciones. Además irá consiguiendo monedas, que el jugador podrá utilizar para comprar accesorios en la tienda del poblado.

3.1.2. Fracciones

La forma de aprender el concepto de fracciones es a partir de misiones. Estas misiones requieren entender las fracciones para poder completarlas exitosamente.

Para completar una misión el jugador debe obtener una serie de componentes.

Por ejemplo, en el juego original existe una misión que consiste en alimentar a la gente del pueblo plantando alimentos en un huerto. Para llevar a cabo esta misión el jugador necesita recoger las semillas de lo que va a plantar. Así mismo



Figura 6: Fracsland: Pueblo

también necesita encontrar diferentes fracciones, ya sea en formato numérico o en representación de porciones. Una vez dispone en su inventario de estos componentes, puede proceder a plantar la porción de alimentos que se requiera. Para hacer esto es necesario seleccionar en el inventario que fracción que completa la porción necesaria.

La forma de obtener las fracciones que el jugador deberá usar, es siguiendo los siguientes pasos:

1. Buscar los altares de fracciones que se encuentran repartidos por el mapa de cada zona.
2. Cuando el jugador encuentra un altar, puede interactuar con él, de esta forma aprenderá nuevas fracciones (*fig: 7*)a.
3. Una vez ha aprendido nuevas fracciones, éstas aparecerán en el inventario del jugador (*fig: 7*b).
4. Cuando el jugador tiene en el inventario las fracciones, puede usarlas para completar la misión correspondiente.



Figura 7: Fracsland: Altar de fracciones y fracciones en inventario

3.1.3. Zonas de Fracsland

El juego está dividido en 4 zonas, de izquierda a derecha, Frozen Woods, Little Beach, FracTown y FracFarm.

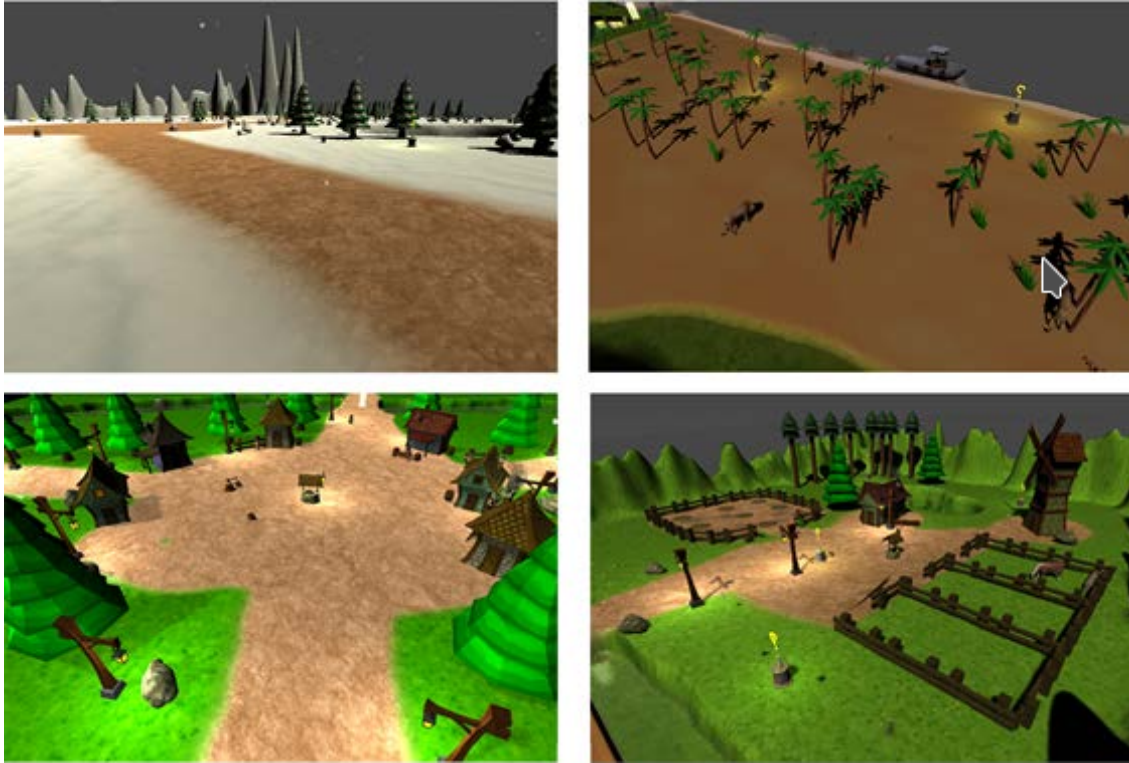


Figura 8: Fracsland: Zones

Frozen Woods: Es una zona que se encuentra nevada, ya que representa que es una zona de montaña. Los principales elementos que se encuentran son árboles. Además existen una serie de NPCs, como por ejemplo: lobos, que irán atacando al jugador. Esta zona está pensada para los jugadores que desean más acción. Es una zona que contiene muchos enemigos y no paran de aparecer nuevos.

Little Beach: Esa área consta de una playa con palmeras y algunos elementos embarrancados en la costa, además de la balsa que permitirá salir de la isla. Asimismo existen unos pocos NPCs, estos NPCs son leones (*fig:9*). Esta es un área más tranquila que la anterior.



Figura 9: Fracslant: Lion

FracTown: En esta área se encuentran los habitantes del pueblo. Eso significa que el jugador se encontrará con las casas de los lugareños y un pozo para suministrar agua. Aquí se encuentra Lord Barus que irá facilitando nuevas misiones a medida que se vayan completando y entre la gente del pueblo hay un mercader que permite vender y comprar objetos.

FracFarm: Esta es la zona más tranquila de todas: no hay ningún enemigo. En esta zona es una granja que permite alimentar a la gente del pueblo, así cómo se puede alimentar a unos conejos, pollos o una vaca. Ya que consta de un huerto para poder plantar alimentos y de esta forma alimentar a las personas y los animales de la granja.

3.1.4. Misiones o retos

En el juego existen diferentes tipos de misiones que pueden ser clasificadas según el nivel de acción deseada, de menor a mayor acción:

- La zona de FracFarm cuenta con tres misiones en las que se necesita cultivar ciertos alimentos en el huerto, para posteriormente dar el alimento a los aldeanos, a la vaca o a los conejos (*fig: 10a*).
- En Little Beach se puede encontrar una misión que consta de reparar un puente derribado por la marea (*fig: 10b*). Para reparar el puente primero se debe buscar madera, la madera se obtiene cortando árboles. Cuando el jugador dispone de la madera necesaria podrá reparar el puente y cruzar al otro lado.

- Por último para los más osados, existe el área de Frozen Woods, donde hay una misión que trata de matar al capitán John. Para matarlo se necesitan fracciones y seleccionar la que acabe con él.



Figura 10: Fracsland: Reto del huerto y del puente

Una vez terminada una misión el jugador ha de volver con Lord Barus para recoger las recompensas y recibir una misión nueva.

3.1.5. Componentes de las misiones o retos

En una misión o reto intervienen diferentes componentes, algunos de estos componentes afectan al jugador, otros en cambio son internos del juego. Los componentes son:

- Objetivo de la misión que el usuario debe completar (esta es la misión que da Lord Barus).
- Objeto con el que el usuario podrá interactuar para realizar la misión. Por ejemplo, en el reto del puente, el objeto de la misión es construir el propio puente.
- Las fracciones que se han ido cogiendo por el mapa, para poder completar dicha misión. Las fracciones se obtienen de los altares que el jugador va encontrando por la isla.

- Materiales necesarios para completar la misión. Por ejemplo, en el puente, toda la madera recogida, que se obtiene cortando árboles. Para poder cortar los árboles, el jugador necesita disponer de un hacha que le permita cortarlos.
- Interfaz gráfica (*UI* o *GUI*) necesaria para poder seleccionar tanto las fracciones como los materiales y herramientas.
- Las misiones tienen efectos especiales que se activan al terminar el reto, bien puede ser al ganar o al perder. Por ejemplo: en el puente, si el jugador no selecciona bien qué fracción completa el puente, al intentar cruzar, se caerá al agua. El efecto en este caso, es la animación de caer al agua.

3.2. ¿Qué mejorar?

Como ya se ha mencionado, para mejorar el juego existente, se ha pensado en añadir nuevos retos creados por los propios jugadores aumentando la jugabilidad e incrementando su vida de uso. Para ello se ha optado por la posibilidad de crear un mod que permitan añadir contenido.

Respecto al contenido que se puede modificar, se ha considerado dar dicho contenido ya creado y que el jugador decida qué contenido poner en cada caso, de entre los disponibles. De esta forma el jugador no requiere conocimientos técnicos, simplemente debe interactuar con el juego y la escena.

Después de analizar detenidamente el juego, se ha considerado que los contenidos a modificar son: Seleccionar un terreno, seleccionar un misión, recursos necesarios para la misión y sus efectos especiales, elementos decorativos para la escena, añadir NPCs, modificar el cielo de la escena y modificar ciertas texturas de ciertos elementos.

A continuación se van a detallar las modificaciones que se van a permitir de cada uno de estos elementos.

3.2.1. Mapa

Por un lado están todos los componentes directamente relacionados con la escena. Son todos aquellos componentes que dan un valor añadido a la escena de forma visual.

En relación al mapa o escenario, el mod va permitir modificar, añadir y editar distintos componentes:

- Seleccionar un tipo de terreno (por ejemplo: playa, huerto, etc.) para establecer

la base del escenario. Este componente es la base sobre el que trabajarán los demás componentes.

- Añadir elementos decorativos, como por ejemplo: edificios, árboles, rocas, etc. El propósito es que el escenario gane estética y visualmente a la hora de jugar en él.
- Insertar NPCs, ya sean humanoides y/o animales. Estos NPCs pueden ser enemigos o amigos, dependiendo del caso.
- Cambiar el cielo de la escena. Con este apartado se pretende dar el efecto de noche, día o tormenta en la escena.
- Modificar las texturas de los elementos decorativos o de misión. Con esto, se pretende dar al jugador la posibilidad de que un mismo objeto pueda tener más de un color o textura.

3.2.2. Misiones

Un reto o misión está compuesto por *Work objects*. Los Work Objects son los objetos necesarios para poder desarrollar un reto. Estos no incluyen el terreno, que está considerado parte del mapa.

Por ejemplo: En el juego original existe una misión que consiste en terminar de construir un puente. Para poder construir el puente es necesario conseguir tablas de madera, para conseguirlas se necesita talar algunos árboles. Una vez conseguida la madera (y teniendo en el inventario las fracciones tal y como se ha explicado en el apartado 3.1.2) el jugador puede completar el reto. Al completarlo se activa un efecto que indica si ha ganado o ha fracasado.

Así pues, se han considerado los siguientes componentes como contenido modificable, además de esenciales para las misiones:

- Seleccionar que misión tendrá el mapa. Esta es la base del mod, pues es el que determina qué objetivos tendrá un mapa en cuestión. Por ejemplo, el reto de reparar el puente.
- Añadir los recursos necesarios para llevar a cabo el reto. Siguiendo con el ejemplo del puente, los árboles necesarios para conseguir la madera, que serán las tablas que repararán el puente.
- Seleccionar qué efectos tendrá el evento cuando el jugador pierda o gane. Por ejemplo, si hace bien el reto, se lanzan cohetes que felicitan al jugador.

3.3. Elementos editables y no editables

Los mapas y objetos, recursos y efectos especiales se incluyen en el escenario. Es decir, el escenario es el conjunto de componentes gráficos que se verán cuando la escena esté lista para jugar, una vez el jugador haya creado el mapa. Los componentes se han dividido en jugables y no jugables.

3.3.1. Elementos editables

A continuación se listan todos los elementos que el jugador podrá usar o modificar para crear la escena.

Se ha dividido los elementos en obligatorios y opcionales.

3.3.1.1. Elementos obligatorios (Work Objects)

Los elementos obligatorios son todos aquellos que corresponden al desarrollo del reto o misión.

Como elemento obligatorio (elemento de Work Object) se ha añadido el terreno, ya que como se va a explicar a continuación sin este no se puede desarrollar la escena.

- El Terreno (*Terrain*) es el mapa que formará la base de la escena, a partir del cual se podrán añadir nuevos elementos. De este elemento lo que se permite modificar es la selección de un terreno entre los disponibles en la interficie de selección de terreno.
- Los Objetivos (*Objectives*) corresponden a las misiones que se deberán completar una vez creado el escenario. La parte modificable sería la posición en el mapa, rotar¹ el elemento y modificar su textura. Como se ha mencionado anteriormente, sería por ejemplo la selección de crear el reto de reparar el puente.
- Los Recursos (*Resources*) son los elementos que se necesitarán para completar la misión, es decir, los prerequisites para la misma. Se permitiría modificar lo mismo que los objetivos: posición, rotación y texturas.
- Los Efectos especiales o visuales (*Effects*) que tendrá el evento al completarse. En este caso la parte editable es la posición en el mapa.

¹Se entiende como rotar el elemento, a girarlo sobre el eje Y, es decir, se gira alrededor de un eje perpendicular a la superficie sobre la cual está el objeto.

3.3.1.2. Elementos opcionales

Estos elementos corresponden a la interacción con la escena, es decir, añaden un valor visual a la escena. A excepción del terreno, que como se ha comentado anteriormente su uso es obligatorio.

- Todos aquellos Elementos decorativos (*Items*) que habrá en la escena. Estos elementos obtienen la misma capacidad de modificación que los objetivos y recursos.
- Todos los Personajes no jugables (*NPCs*) que tendrá la escena, esto incluye personajes de tipo humanoide y animales. Se permite editar la posición del mapa y rotar.
- El cielo (*Sky*) es el que dará el efecto de día/noche en el escenario.
- Los objetivos, recursos y elementos decorativos disponen de la posibilidad de modificar sus texturas (*Materials*).

3.3.2. Elementos no editables

Los elementos no editables son aquellos que el usuario no tiene acceso de forma directa, si se modifican es de forma indirecta sin que el jugador intervenga.

- Las luces que va a tener la escena. Estos son los componentes que iluminan la escena, para que no esté a oscuras.
- Las cámaras son aquellos elementos que sirven para que el jugador vea la escena. Las cámaras son estáticas en el modo diseño. Cuando el usuario entra en modo probar escena, la cámara se modifica automáticamente para situarla en la posición adecuada para poder jugar. El jugador no puede tocar ningún elemento de las cámaras, por ello se considera como elemento no editable.
- Eventos del sistema, son aquellos elementos que van a estar presentes en la UI. Por ejemplo botones, textos, etc. Lo que significa que no serán editables ni configurables desde el mod.

3.4. ¿Cómo mejorar el juego?

Puesto que este proyecto es una aplicación para generar mods, se ha tenido que diseñar una interfaz que pueda cumplir con los requisitos. Es decir, que sea amigable,

intuitiva y de fácil uso por jugadores de 10-12 años, todo esto sin olvidar que se debe poder elegir entre diferentes objetos con los que interactuar.

Para ello se ha pensado un diseño de interfaz de usuario que permita ver qué elemento se selecciona para insertar en el mapa. Asimismo, también se ha de poder ver dónde quedará el objeto, ya que hay que indicar en qué parte del mapa se quiere insertar este objeto.

El diseño que se ha diseñado consta de 3 vistas distintas principales, de las cuales dos se heredarán para que con mínimas modificaciones se adapten a todas las necesidades. Las vistas son: menú principal, menú de Work Objects, menú lateral y cambio de nombre.

A continuación se exponen las diferentes vistas que se han pensado, para facilitar la explicación se han añadido colores y una numeración (ver figura 11).

3.4.1. Menú principal

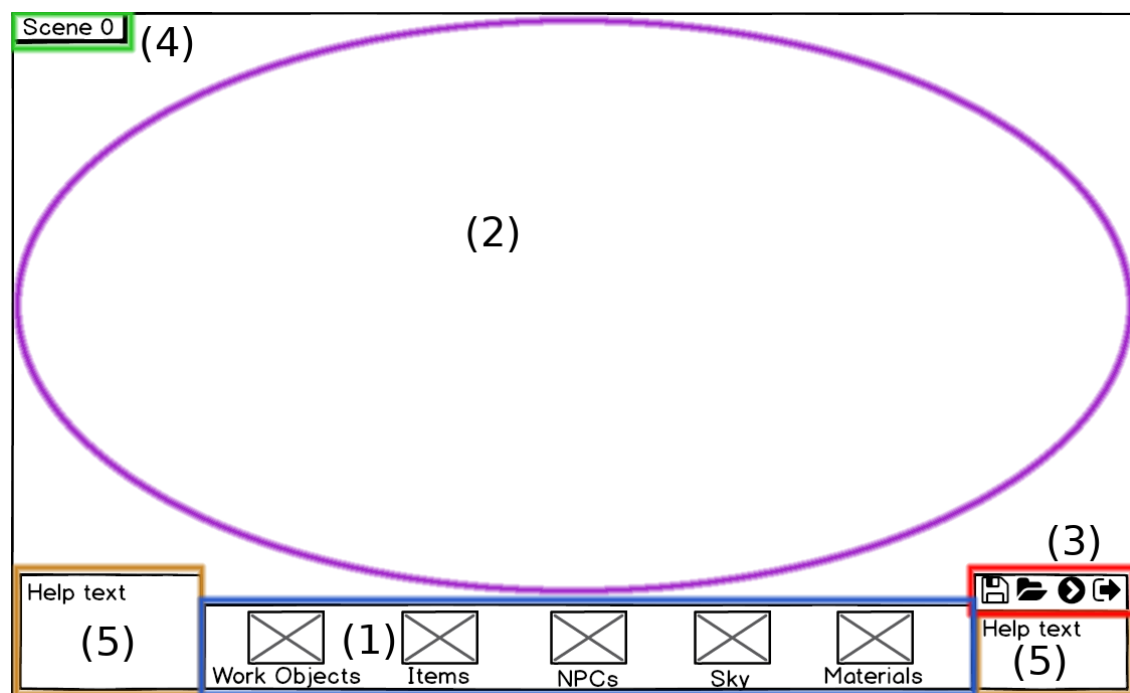


Figura 11: Mockup Menú inicial

La UI principal se ha dividido en 5 partes, diferenciadas en la figura 11. Estas partes constan de:

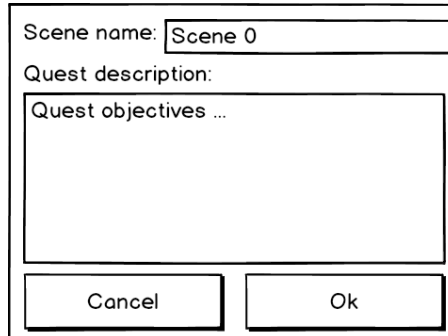
1. En la figura 11 se resalta en color azul (número (1)) la sección de la GUI que corresponde a la parte principal del menú, en este sector se concentran los

botones necesarios para modificar la escena.

- Work Objects: Corresponde a la edición de elementos relacionados con el reto, véase: *Terrain*, *Objectives*, *Resources* y *Effects*. A su vez dichos elementos son de carácter obligatorio, a diferencia del resto que son opcionales.
 - Items: Es la parte encargada de insertar elementos decorativos en la escena, estos elementos aportan un valor visual añadido, ya que terminan de completar la escena para que no se vea vacía.
 - NPCs: Permite insertar personajes no jugables a la escena. Estos personajes pueden ser de aspecto humanoide o animal.
 - Sky: En este apartado se da la posibilidad, al jugador, de dar al escenario el efecto de noche, día o tormenta. Se permite escoger un tipo de clima de entre los disponibles.
 - Materials: En esta sección se da la posibilidad de modificar las texturas de ciertos elementos (ver la sección del escenario), los elementos son: Objetivos de la misión (*Objectives*), Recursos necesarios (*Resources*) y Elementos decorativos (*Items*).
2. En la figura anterior se resalta en color morado (número (2)) la parte de la GUI que corresponde al área donde se irá viendo y modificando la escena, es decir: se irán viendo el terreno, los objetivos, los elementos decorativos, etc. Esta zona estará presente en todas las interfaces diseñadas y para facilitar el uso estará siempre ubicada en el mismo sitio.
 3. Siguiendo con la figura anterior (*fig: 11*), se resalta en color rojo (número (3)) el área donde se encuentran los botones de configuración. Estos botones son: Guardar escena, cargar escena, ejecutar la escena para probarla y salir del editor.
 4. La sección resaltada en color verde y con el número 4 representa el apartado donde aparece el nombre de la escena que se esta creando/modificando. Para facilitar el uso al jugador, esta sección aparece en todas las interfaces pensadas, y aparece siempre en la misma posición.

Si se desea modificar el nombre, es posible hacer click encima del nombre de la escena. Al presionar el botón, aparece una ventana (*fig: 12*), dicha ventana permite modificar el nombre y definir el objetivo de la misión que se esta desarrollando. La definición del objetivo es el texto que aparecerá en el juego, en otras palabras: es la descripción de la misión que dará Lord Barus.

5. En la figura 11 se resalta en color marrón (número (5)) las zonas donde se mostrarán textos de ayuda, en caso de ser necesario.



A dialog box titled 'Scene name: Scene 0'. It contains a text input field with 'Scene 0' inside. Below it is a label 'Quest description:' followed by a large text area containing 'Quest objectives ...'. At the bottom are two buttons: 'Cancel' and 'Ok'.

Figura 12: Mockup Cambio de nombre

3.4.2. Menú de *Work Objects*

Como se acaba de ver, el botón de *Work Objects* (representado en la zona del número 1 y de color azul en la figura 11) es el encargado de gestionar como se comportará el reto. La interfaz pensada, es similar a la principal, pero modificando la sección principal, la que se encarga de la gestión de la escena (ver figura 13).

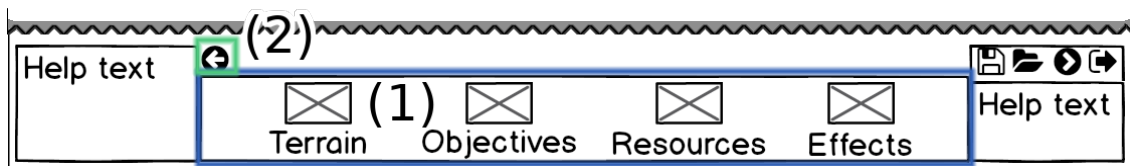


Figura 13: Mockup Menú *Work Objects*

Estos son los cambios que tiene esta interfaz, respecto a la principal:

1. En la figura 13 se resalta en color azul (número (1)) la sección de la GUI que se encarga de modificar los elementos relacionados con el reto, además estos elementos son obligatorios, es necesario como mínimo un objeto de cada uno de estos elementos.
 - Terrain: Permite seleccionar un terreno, este terreno será la base del juego. Sin este elemento no se puede insertar ningún otro elemento.
 - Objectives: Aquí se selecciona cuál o cuáles son los objetivos que se deberán completar. Es la selección de la misión que se quiere realizar. Por ejemplo, seleccionar el reto de plantar comida en el huerto.

- Resources: En este punto se definen los materiales necesarios para completar el reto, son los prerequisites. Siguiendo con el ejemplo anterior, seleccionar y distribuir por el mapa algunas zanahorias para poder cultivarlas.
 - Effects: Se definen los efectos que pueda tener el reto, por ejemplo, unos cohetes para celebrar que se ha completado el reto correctamente.
2. En la figura anterior se resalta en color verde y número 2 el área donde se ha añadido un botón de retroceso que permite, al jugador, volver al menú principal.

3.4.3. Menú lateral

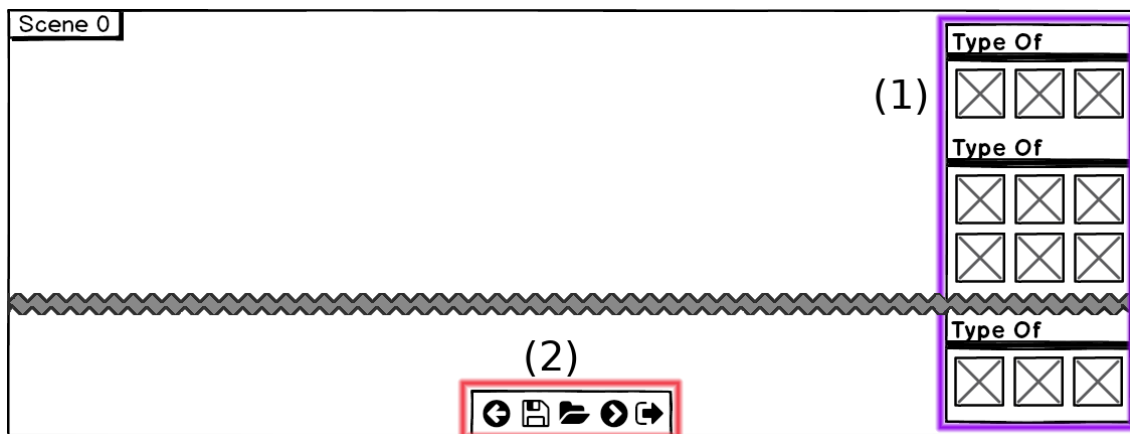


Figura 14: Mockup Menú lateral

Para facilitar la inserción de elementos a la escena, se ha pensado un menú lateral. Este menú se ha pensado así para que la zona del terreno sea más grande y así tener un mayor control. Este diseño mantiene la zona del terreno y el cambio de nombre, y añade dos zonas nuevas¹⁴. Estas zonas son:

1. En la figura 14 se resalta en color morado (número (1)) la sección de la GUI que corresponde a donde estarán los elementos que se quieran añadir a la escena, dependiendo de lo que se quiera añadir habrá unos elementos u otros. Por ejemplo: si se desea insertar un terreno, aquí saldrán los terrenos disponibles. Estos terrenos estarían clasificados según el tipo de terreno, playa, huerto, etc.
2. En la figura anterior se muestra en color rojo y con el número 2 los botones de configuración, explicados en el menú principal, trasladados a la parte inferior de la interfaz y centrados. Además se ha añadido otro botón, el que permite volver al menú anterior.

4. Diseño

4.1. Draft de la interficie y su funcionamiento

4.1.1. Flujo de la UI

Como se ha visto en la sección anterior hay un menú principal desde donde se puede pasar al menú de Work Objects, menú con el contenido obligatorio, y/o al menú lateral, menú con los distintos diseños de un elemento. El diagrama de flujo de la UI, sería:

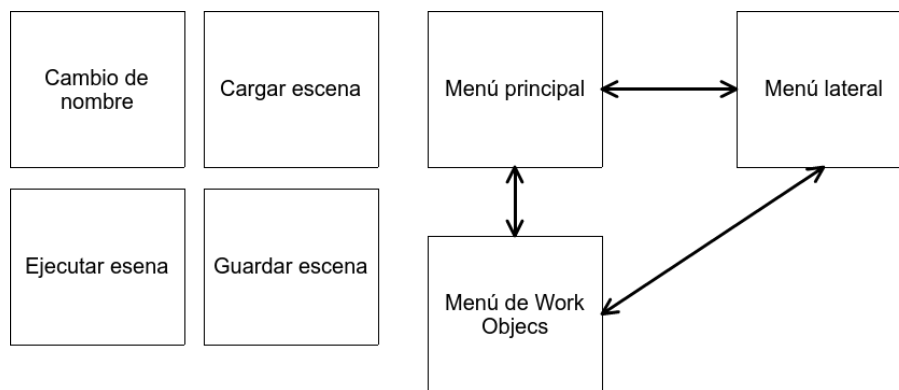


Figura 15: Diagrama de secuencia de la UI

Tal y como se observa en la figura anterior, existen botones que son accesibles en cualquier momento durante la ejecución del juego. Estos botones son: Cambiar el nombre de la escena, cargar y guardar una escena y ejecutar la escena que se esté modificando.

Por otro lado existen elementos de la interfaz que son accesibles a partir de otros elementos. El menú principal puede abrir el menú lateral y así elegir que elemento añadir a la escena. Al volver hacia atrás, se volverá al menú principal.

De la misma manera el menú principal puede abrir el menú de Work Objects y este abrir el menú lateral. Al volver a tras, primero se volverá al menú de Work Objects y posteriormente al menú principal.

4.1.2. Casos de uso

En este proyecto existe un solo un actor implicado, el jugador. Para facilitar la comprensión de los casos de uso se han agrupado por colo y numeración (*fig:16*). A continuación se explican los diferentes casos de uso, según los grupos a los que pertenecen:

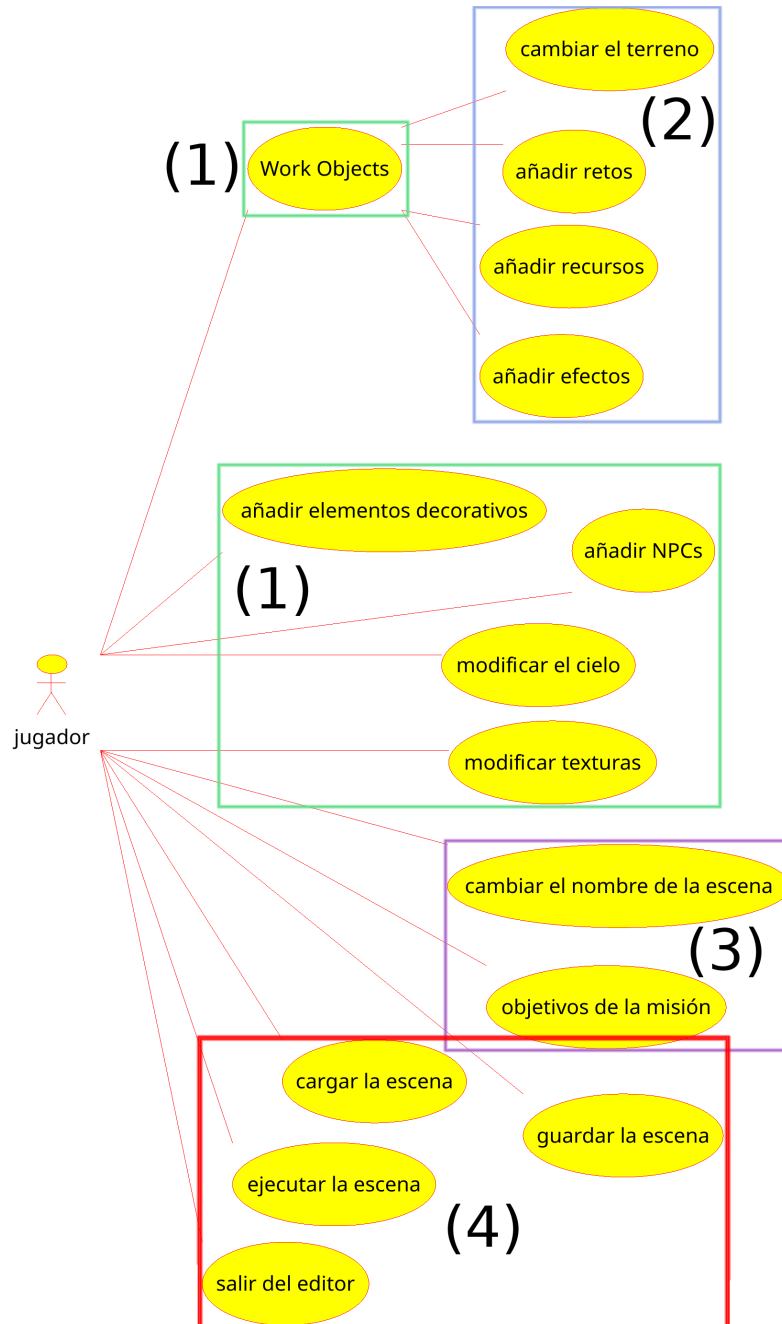


Figura 16: Diagrama de casos de uso

4.1.2.1. Primer grupo (Color verde)

Este primer grupo (número 1 en la figura 16) corresponde al menú principal (definido con el color azul y número 1 en la figura 11), es donde se encuentran los elementos para modificar la escena.

- **UC1: Menú de Work Objects:** El jugador puede acceder al menú encar-

gado de gestionar el reto, para de esta manera poder crear un nuevo reto en la escena. Es decir el jugador puede acceder al menú de Work Objects (visto en la figura 15).

- **UC2: Añadir elementos decorativos (*Items*):** El usuario puede añadir elementos decorativos por la escena. Estos elementos pueden ser edificios, árboles, rocas, etc. Todos los elementos decorativos ya están creados, el usuario final solo debe escoger cuáles quiere usar.

Para posicionar un elemento en el mapa, la persona, selecciona el icono del elemento que desea insertar desde el menú lateral, y al mover el ratón por el mapa el objeto se irá desplazando ahí donde apunte el ratón. Finalmente para posicionarlo, el jugador ha de presionar el ratón y el elemento quedará fijado en dicha posición.

- **UC3: Añadir *NPCs*:** El jugador puede añadir personajes no jugadores, ya sean humanoides o animales por el mapa. El jugador decide si quiere que sean enemigos o amigos.

Comparte método de inserción en el mapa con el caso de uso anterior.

- **UC4: Cambiar el cielo (*Sky*):** Si se desea se puede cambiar la textura que define el cielo de la escena, para dar efecto noche/día.

La forma de inserción consiste en: existen una serie de iconos en el menú lateral que representan los distintos tipos de cielo. Al presionar uno, el cielo cambia automáticamente al tipo de cielo seleccionado.

- **UC5: Modificar algunas texturas (*Materials*):** Es posible modificar ciertas texturas del juego, para dar un aspecto nuevo a un objeto ya existente. Se pueden modificar las texturas de los elementos de tipo Objective, Resource e Item.

Para modificar una textura se selecciona la textura que se desea, de entre las disponibles en el menú lateral, y con el ratón se hace click encima del elemento que se desee modificar.

4.1.2.2. Segundo grupo (Color azul)

Este apartado corresponde a los elementos del menú de Work Objects, es decir los encargados de gestionar los retos. Se accede a partir el UC1.

- **UC6: Cambiar el terreno (*Terrain*):** El jugador puede elegir un terreno de entre los que hay disponibles. De esta forma puede comenzar a definir como será el mapa final. Si elige un mapa con playa, este hecho comenzará a condicionar las subsiguientes elecciones, si se desea mantener una concordancia en la escena.

Para seleccionar el mapa, el usuario presiona el icono que representa un mapa en concreto desde el menú lateral, y este aparece en el escenario. Si selecciona otro mapa, el anterior desaparecerá.

- **UC7: Añadir objetivos (*Objectives*):** El usuario puede decidir cuál o cuáles serán los retos que se deberán completar para llevar a cabo, una vez el mapa esté terminado. En otras palabras, el usuario, puede elegir, de entre los disponibles, qué retos tendrá esta escena. Como por ejemplo, reparar un puente.

La forma de añadir un objetivo es similar a los casos de uso UC2 y UC3.

- **UC8: Añadir recursos (*Resources*):** El jugador puede añadir al mapa todos aquellos componentes que crea necesarios para completar el reto. Como árboles para conseguir madera, zanahorias para poder plantarlas, etc.

La forma de posicionar el recurso en el mapa es idéntica a insertar elementos decorativos, NPCs y objetivos.

- **UC9: Añadir efectos (*Effects*):** Cuando un jugador completa un reto, tanto si gana o pierde, este tendrá un efecto visual, especial o una animación que indique al jugador si ha ganado o perdido. Por ejemplo en Fracslan, al completar el reto del puente, si se hace mal, el usuario cae al agua.

La forma de insertar los efectos en la escena es la misma que en los casos vistos anteriormente.

4.1.2.3. Tercer grupo (Color morado)

Este grupo de casos de uso se encargan de la configuración de la escena, es posible acceder a ellos en cualquier momento.

- **UC10: Cambiar el nombre de escena:** La persona que está modificando o creando la escena, puede asignar a la escena el nombre que desee. Este nombre es el nombre de zona que aparecerá posteriormente en Fracslan.

- **UC11: Objetivos de la misión:** El usuario que esté jugando a Fracslan debe tener información sobre la que misión debe realizar. Esa información es asignada al crear la escena y la asigna la persona que la crea.

4.1.2.4. Cuarto grupo (Color rojo)

Este conjunto de casos de uso se encargan de la configuración global del juego, es posible acceder a ellos en cualquier momento.

- **UC12: Guardar la escena:** El jugador puede guardar en cualquier momento la escena, para retomarla posteriormente. O para que Fracslan pueda usar dicha escena.
- **UC13: Cargar una escena:** Puede cargar una escena ya guardada. Y de esta forma continuar con el proceso de creación de la escena.
- **UC14: Ejecutar la escena:** En cualquier momento el usuario puede decir probar la escena, ver como quedará cuando se ejecute en Fracslan.
- **UC15: Salir del editor:** En cualquier momento es posible salir del editor, cerrando así el programa.

4.2. Diagrama de dominio

Tal y como se ha diseñado (ver la figura 17), la clase que gestiona al resto es la clase *Scene*, ya que se crea al comenzar el juego. De esta clase están relacionadas directamente algunas clases que también se generan al iniciar el juego, como el *GameController*, el *Terrain*, etc. Estas clases que se relacionan directamente con la escena, son las clases más importantes del proyecto, pues se encargan de gestionar el juego de una u otra manera y se crean al inicializar el juego. De entre estas clases hay alguna que es interna de Unity, en el diagrama (*fig: 17*) están expresadas con la palabra Unity entre paréntesis.

La figura 17) resume el diagrama del modelo de dominio diseñado para este proyecto.

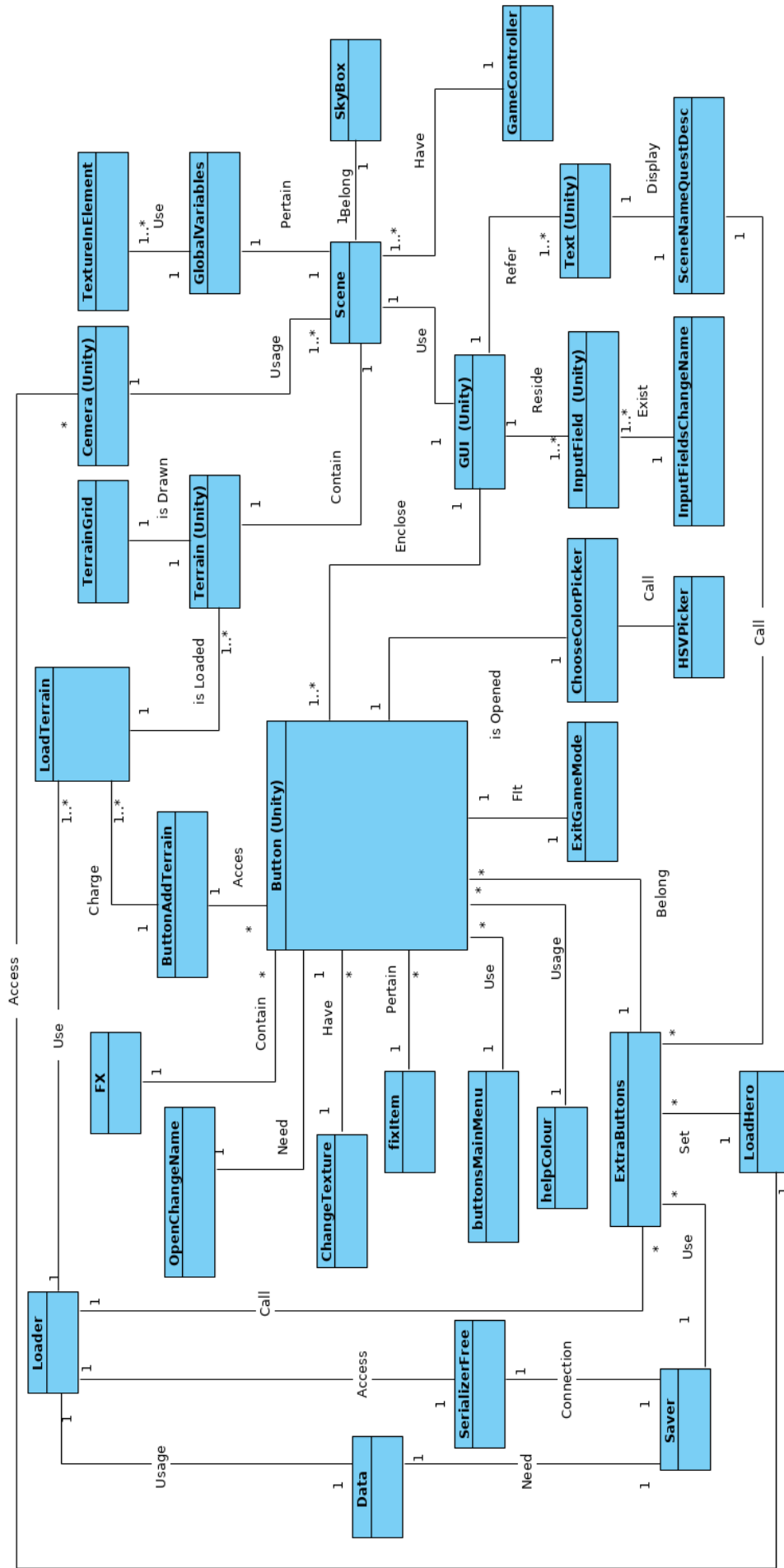


Figura 17: Diagrama de modelo de dominio

Se ha implementado el GameController como una instancia de patrón Singleton. Esta clase se encarga de ciertas funcionalidades, por ejemplo mostrar/ocultar diferentes componentes de la UI. Muchos de los botones existentes deben cambiar la UI; la única forma de hacerlo es a través del GameController.

La forma de acceder al GameController es un poco complicada, debido a que esta clase se inicializa al comenzar el juego y no puede volver a crearse. La forma de poder llamar a sus funcionalidades desde otras clases es siguiendo estos pasos:

1. Desde la clase que se necesita usar el GameController, se debe hacer una búsqueda de la instancia.
2. Una vez encontrada la instancia, ya es posible usar sus funcionalidades.

Debido a la complejidad del diagrama se ha extraído a otro diagrama las relaciones con el GameController (*fig:18*).

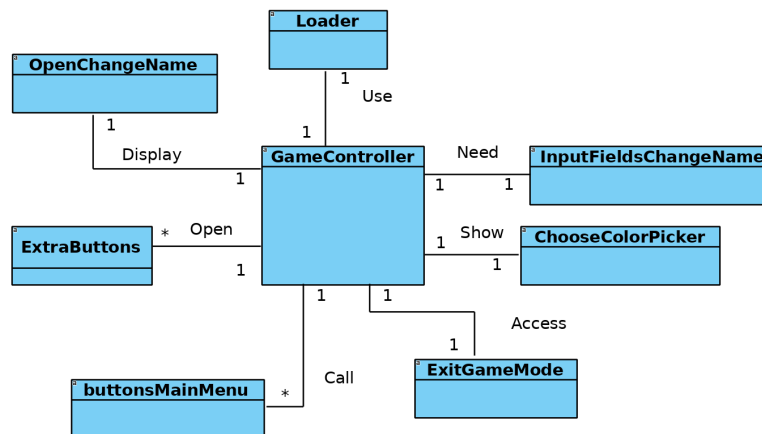


Figura 18: Diagrama de casos de uso del *GameController*

Debido a como se implementa Unity internamente ha surgido un problema a la hora de transmitir información entre clases. El método que se ha elegido para transmitir información entre clases, ha sido con la creación de una clase estática con variables también estáticas, teniendo en cuenta las limitaciones de Unity no había solución realmente buena.

Esta clase es *GlobalVariables* y es usada por el resto de clases. Las variables gestionan los estados del mod y de la interficie. En el diagrama está relacionada directamente con la escena, pero realmente todas las clases acceden a ella, ya que es estática. Eso quiere decir que aunque no se haya dibujado una relación con todas las demás clases, no quiere decir que no exista, solo que no se ha dibujado para

no complicar el diagrama con demasiadas líneas que convergen en esta clase. La relación que tienen las clases con esta es:

$$GlobalVariables^1 -^* OtrasClases$$

Dicha clase sirve para transferir información entre ellas. Lo normal, cuando se necesita mover información, es crear una instancia de la clase a la que va dirigida y pasarle la información. En este caso no siempre es posible, porque la clase a la que va dirigida ya está creada por otro evento del sistema. Otra posibilidad es enviarle directamente eventos a la otra clase, pero entonces se complica la gestión del juego porque aparecen problemas de sincronización. Mediante la clase *GlobalVariables*, se soluciona el problema sin tener que transmitir, recalcular o crear clases adicionales.

Como se puede ver en el diagrama (*fig:17*), existen una serie de clases relacionadas directamente con *Button (Unity)*, esto es debido a que estas clases se encargan de gestionar ciertos botones. Y para poder gestionarlos deben poder acceder al *Listener* del botón.

Con el terreno intervienen dos clases, una que se encarga de cargar el terreno seleccionado por el jugador o el terreno guardado, al guardar la escena. Y otra clase que se encarga de dibujar en el terreno una cuadrícula que indica al jugador si puede o no insertar un objeto en una posición determinada.

Por otro lado se usarán un par de clases externas: *HSVPicker* y *SerializerFree*. La primera se usa para poder mostrar un selector de color y la segunda para Serializar/Deserializar los componentes de la escena para poder guardar y cargar la partida.

4.3. Insertar elementos en el escenario

4.3.1. Funcionamiento general

Para poder insertar un elemento en la escena se requieren varios elementos e interacciones.

1. En la escena debe de estar cargado un terreno, sin el terreno no es posible añadir ningún elemento a la escena.
2. Una vez colocado el terreno, se puede seleccionar el elemento que se desea insertar, por ejemplo un árbol.
3. Con el elemento seleccionado al mover el ratón por el mapa, el objeto aparece ahí donde esté el puntero del ratón, y se puede desplazar.

4. Finalmente, haciendo click, el objeto se queda fijado en la posición indicada.

4.3.2. Estrategia utilizada

A continuación se va a exponer cómo aparece el objeto en el mapa, así cómo queda fijado.

Una vez seleccionado el objeto a insertar, se crea un RayCast desde la cámara hacia el terreno. El RayCast crea un rayo invisible desde la cámara hacia la dirección que se le indique (en este caso hacia la posición del ratón), y devuelve la posición del primer objeto que se encuentra o el objeto del tipo que se requiere; en este caso se quiere obtener el terreno, así que de momento se ignoran los demás tipos de objetos.

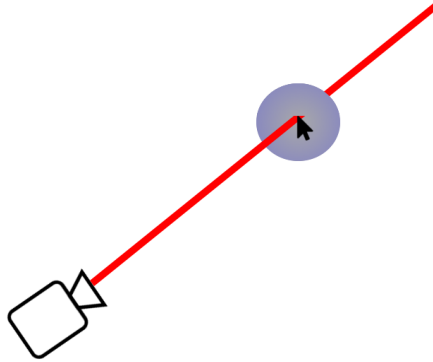


Figura 19: Demostración del RayCast

Una vez que se tiene la posición del terreno, el objeto se mostrará en esa posición, cuando se mueva el ratón, el objeto también lo hará. El objeto se irá desplazando porque se irán lanzando RayCast contra el puntero del ratón y de esta forma se puede saber la nueva posición en el mapa. Con la nueva posición se hace una traslación del objeto. Este proceso se repite hasta fijar el objeto. De esta forma se consigue simular el *Drag & Drop*.

Para poder fijar un objeto en una posición, se comprueba si el objeto a insertar choca con otro objeto. La forma de saber si existe un objeto en el área de acción del elemento es lanzando otro RayCast, en este caso contra los elementos que ya hay en el mapa. Si no choca con nada, se verá una cuadrícula verde, en caso contrario, en el punto donde choque se verá en rojo, impidiendo la inserción.

Si la posición a insertar es correcta, pero se desea que el objeto apunte hacia otro lado, se puede rotar el objeto 90° con el botón derecho del ratón o *Shift* + botón derecho del ratón.

Todo este proceso se repite constantemente, hasta que el jugador decide que una posición es correcta; entonces para que quede fijo en una posición, se debe hacer click con el botón izquierdo del ratón.

Este proceso es idéntico para cualquier elemento que se desee insertar, a excepción de dos casos.

- El primer caso, es insertar un cierto tipo de reto, por ejemplo un puente, que debe ser insertado ahí donde haya un obstáculo a evitar.

Para saber si el reto esta posicionado entre los dos lados del obstáculo, por ejemplo, el borde de un acantilado, se realiza un RayCast, contra el puntero del ratón, para ver si el rayo choca contra ciertos *Colliders*² que se han puesto en el mapa. Estos Colliders determinan la zona del obstáculo, si el objeto esta donde toca, podrá insertarse.

- El segundo caso consiste en el proceso de modificar una textura.

La forma de funcionar es como se ha mencionado, con RayCast, pero en este caso no hay previsualización ni cuadrícula.

No tiene una previsualización puesto que aún no se ha especificado el elemento a modificar. Y no aparece la cuadrícula, porque no es una inserción de objeto, así que no puede haber colisión.

La forma de modificar la textura es una vez seleccionada la textura deseada, se hace click en aquel elemento que se quiera modificar. De esta forma se lanzará un RayCast contra el ratón para buscar el objeto, si choca contra un objeto valido, se modificará la textura. Por ejemplo se desea modificar la textura de una roca: se seleccionará la textura a poner y a continuación se hace click en la roca, si todo ha ido bien, se habrá modificado la textura.

4.4. Herramientas utilizadas

Para poder llevar a cabo este proyecto, se ha utilizado herramientas ya creadas para poder acelerar el desarrollo. Algunas herramientas ya creadas provienen de Fracsland, estas no se mencionan pues se consideran parte del juego. Las herramientas se han dividido en función de cuál es el objetivo que tienen.

²Un Collider es un objeto interno de Unity: este objeto se usa en la escena, ya que es un objeto invisible, y sirve para ayudar a desarrollar la escena. Existen distintos tipos de Colliders: para delimitar un objeto, delimitar zonas, crear posiciones de Respawn, etc. Para más información consultar la documentación oficial.

4.4.1. Work Objects

Para posicionar elementos en el terreno se ha usado un *Script*, creado por René Klačan³. Este script agiliza la comprobación a la hora de insertar elementos en el mapa, tanto para el desarrollador como para el usuario final.

El script funciona lanzando rayos contra el mapa (RayCast), en aquella zona donde se encuentre el ratón. Al lanzar los rayos comprueba si existe algún elemento entre la cámara y el terreno. Si no hay ningún elemento, se mostrará una cuadrícula en verde. Si hay algún elemento, la zona donde hay choque, se vuelve de color rojo. De esta manera el usuario puede ver fácilmente si puede o no puede insertar un objeto en una zona determinada.

Se han añadido una serie de modelos para tener más diseños de misiones, de recursos necesarios para llevar a cabo el reto y de efectos especiales.

- Simple Wooden Bridge⁴: Como diseño de misión se ha añadido este modelo de puente para tener un mayor repertorio de puentes.
- Para poder tener un mayor repertorio de recursos que puedan ser usados, estos recursos también se pueden usar como elementos decorativos, se ha usado una serie de elementos:
 - Stone⁵: Modelo 3D de una roca que puede ser insertada en el mapa.
 - Carrot 3D Model⁶: Se utiliza para usar un modelo 3D de una zanahoria.
 - Tree 3D Model⁷: Modelo 3D de un árbol.
 - Palm 3D Model⁸: Modelo 3D de una palmera.
- Space Shooter tutorial⁹: Del tutorial que se ha realizado para aprender Unity, se ha obtenido el efecto de explosión de cohetes.

4.4.2. Elementos decorativos y NPCs

Con el fin de tener un mayor número de elementos decorativos y NPCs, se ha utilizado algunos diseños ya prediseñados. Los recursos necesarios para la misión,

³<http://rene.klacan.sk/unity3d/games/2015/01/23/draw-grid-on-the-terrain-in-unity/>

⁴<https://www.assetstore.unity3d.com/en/#!/content/819>

⁵<https://www.assetstore.unity3d.com/en/#!/content/651>

⁶<https://archive3d.net/?a=download&id=cdfa296b>

⁷<https://archive3d.net/?a=download&id=28c99091>

⁸<https://archive3d.net/?a=download&id=95c7cf33>

⁹<https://www.assetstore.unity3d.com/en/#!/content/13866>

también se pueden usar como elementos decorativos, como ya se han expuesto anteriormente, no se volverán a exponer.

- Oriental Structure Pack - Lite¹⁰: Diseño de un edificio de estilo oriental en 3D.
- Cute Kitten¹¹: Modelo 3D de un gato, que puede servir como elemento siendo un animal, en principio, inofensivo.
- Animated Spider¹²: Modelo 3D de una araña gigante.
- Goblin¹³: Modelo 3D de un Goblin.
- Wizard Girl¹⁴: Modelo 3D de una maga.

4.4.3. Texturas y colores

Se ha usado un conjunto de texturas que permite aumentar los modelos de texturas del juego. De esta manera el jugador puede modificar la textura de los elementos permitidos. El pack usado es: Surfaces Lite¹⁵.

Para que el usuario puede elegir más texturas, se ha usado un selector de color para la GUI. De esta forma el jugador puede seleccionar un color que desee y usarlo en el elemento que le convenga. El selector de color se ha obtenido de un proyecto para Unity llamado: HSV Color Picker Unity¹⁶.

4.4.4. Guardar y cargar una escena

Para guardar y cargar una escena se necesita Serializar y Deserializar los componentes que forman la escena, para hacer esta labor se ha usado un componente que lo hace automáticamente. Este componente se llama: Serializer Free¹⁷.

Por otro lado, para poder seleccionar donde guardar la escena y que fichero cargar, para continuar editando, se ha necesitado la ayuda de un complemento. Dicho complemento es Unity Standalone File Browser¹⁸ que permite abrir un explorador de archivos desde dentro del juego.

¹⁰<https://www.assetstore.unity3d.com/en/#!/content/81021>

¹¹<https://www.assetstore.unity3d.com/en/#!/content/33121>

¹²<https://www.assetstore.unity3d.com/en/#!/content/22986>

¹³<https://www.assetstore.unity3d.com/en/#!/content/12131>

¹⁴<https://www.assetstore.unity3d.com/en/#!/content/4180>

¹⁵<https://www.assetstore.unity3d.com/en/#!/content/89739>

¹⁶<https://github.com/judah4/HSV-Color-Picker-Unity>

¹⁷<https://www.assetstore.unity3d.com/en/#!/content/84730>

¹⁸<https://github.com/gkngkc/UnityStandaloneFileBrowser>

5. Implementación y simulaciones

5.1. Implementación

5.1.1. Diagrama de clases

Una vez hecho el estudio de los problemas habituales en los juegos, sobre todo en los Serious Games, y habiendo pensado cómo solucionar este problema para mejorar Fracslan, se ha procedido a desarrollar el programa.

Para el desarrollo del proyecto se ha usado el diseño del diagrama de dominio; al finalizar el proyecto ha quedado el diagrama de clases mostrado en la figura 20.

Para no sobrecargar el diagrama, se han puesto todas las clases usadas dentro de un recuadro rosa, esto quiere decir que todas las clases que se encuentran en el recuadro, están relacionadas por un lado con las clases nativas de Unity (visto en el diagrama de dominio). Y por otro lado están relacionadas con la clase estática *GlobalVarialbes*. Eso quiere decir que aquellas clases que no tienen relación dentro del recuadro rosa, en realidad tienen relación solo que no se muestra en el diagrama, porque su relación es con una clase nativa de Unity y/o con la clase estática.

Como se observa en el diagrama de clases, la clase *GameController*, contiene una serie de funciones públicas que serán llamadas desde otras clases, principalmente para gestionar qué GUI se mostrará en cada momento. Ello permite una transición entre interfaces gráficas.

La clase *ExtraButtons* llama a algunas funciones que se encuentran en otras clases, ya que se encarga de gestionar los botones de configuración de la escena, véase: guardar y cargar la escena, ejecutar la escena y salir del editor.

Pero probablemente una de las claves es *GlobalVarialbes*, ya que como se mencionó en el apartado de diseño, es una clase que permite transmitir información entre clases. Sin esta clase muchas de las funcionalidades no podrían funcionar por la falta de la información necesaria.

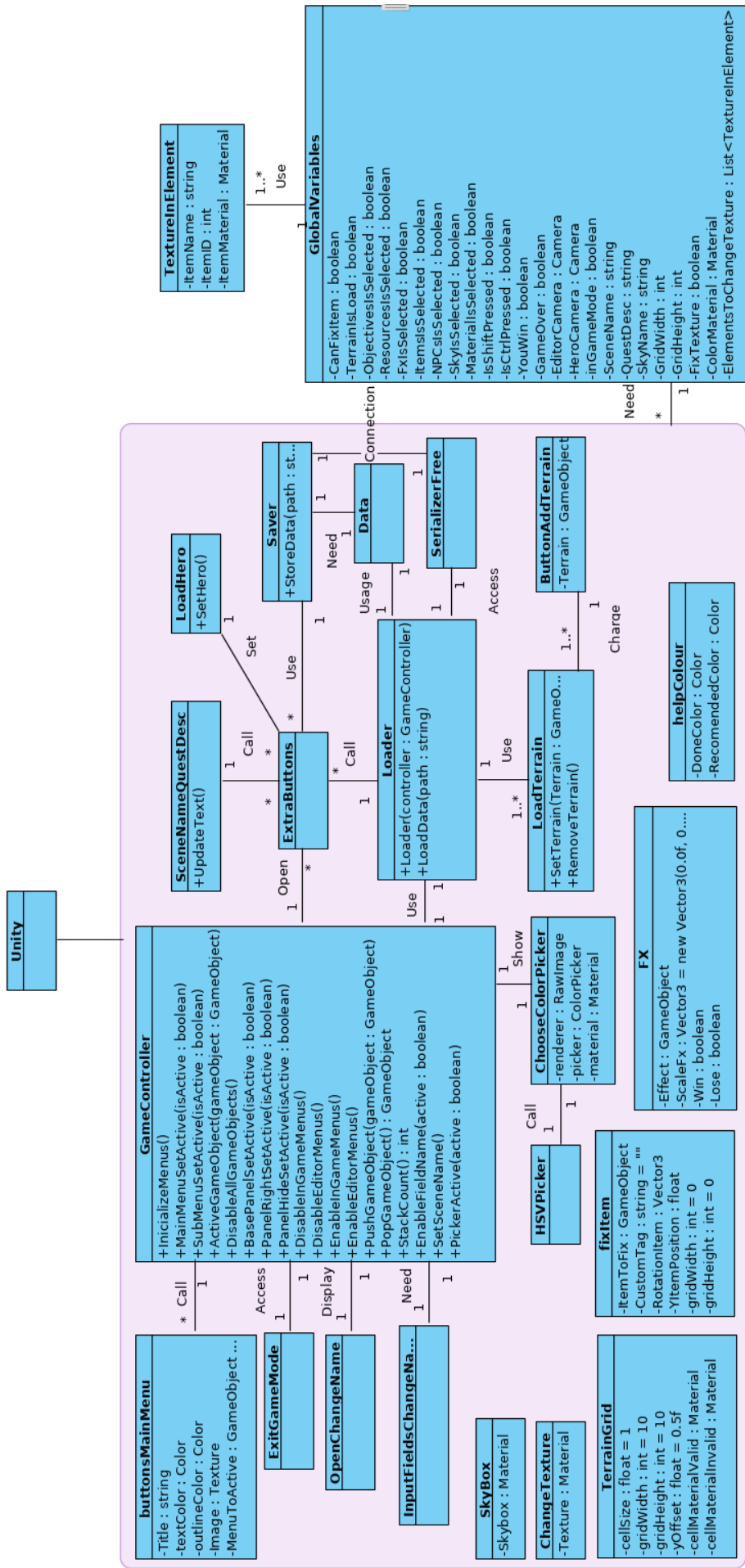


Figura 20: Diagrama de clases

5.2. Funcionamiento del juego

Para desarrollar la interfaz gráfica se han usado los *Mockups* diseñados; a continuación se va a exponer cómo se han desarrollado y su funcionamiento, desde un escenario vacío a probar el escenario finalizado.

5.2.1. Introducción al juego

Cuando se ejecuta el mod aparece un escenario vacío. En la interfaz se encuentran una serie de botones que permiten seleccionar con que elemento trabajar (*fig:21a*). Es decir, si se va a trabajar con los Work Objects o con elementos decorativos de la escena. También permite al usuario cargar un escenario ya creado, desde los botones de configuración.



Figura 21: Menú principal y menú de Work Objects

Para comenzar a crear la escena, el jugador ha de dirigirse a la zona de Work Objects (*fig:21b*), en esta zona se encuentran todos los elementos necesarios para que funcione el reto que se desea crear.

5.2.2. Añadir elementos a la escena

Para poder comenzar la escena es necesario insertar primero un terreno, ya que el juego no permite acceder a otro menú que no sea la inserción del terreno (salvo el menú que permite modificar el nombre de la escena), en el menú de Work Objects se selecciona insertar Terrain. Al entrar en esta sección la interfaz se modificará (*fig:22a*) para facilitar la inserción del terreno.

Para insertar el terreno que se desea, simplemente el usuario debe presionar el botón del mapa que quiere. Una vez presionado, el terreno aparecerá automáticamente en la escena (*fig:22b*).

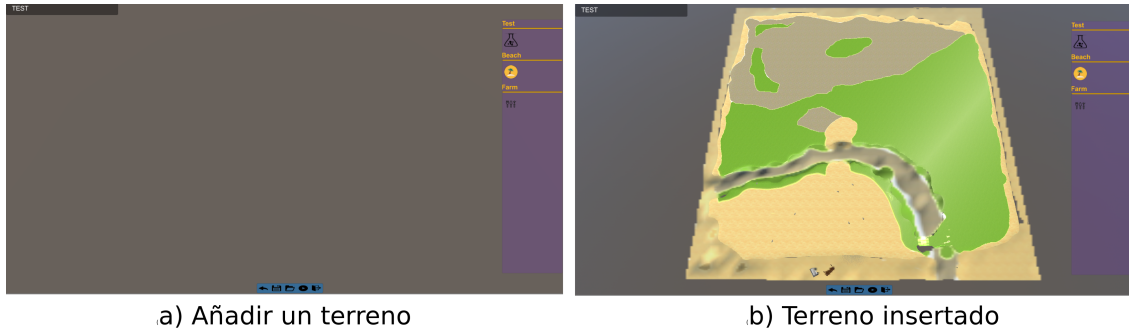


Figura 22: Añadir un terreno y terreno ya insertado

Una vez ya se ha añadido el terreno a la escena es posible insertar más elementos.

Por ejemplo se inserta el reto que tendrá esta escena, para ello, desde el menú de Work Objects, se selecciona el botón de Objectives. Al entrar en este apartado, el jugador, observa una interfaz parecida con la de añadir terreno (*fig:23a*).

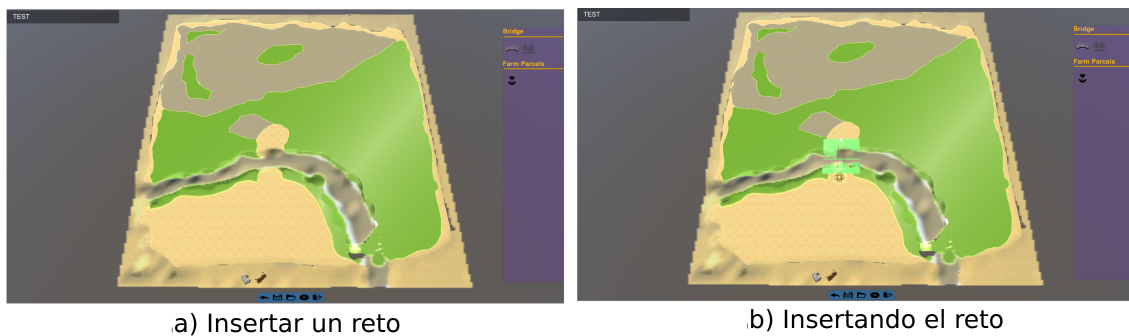


Figura 23: Añadir un reto e insertar el reto

Para insertar un reto, se debe seleccionar el botón que corresponda al reto deseado. Ahora el usuario mueve el ratón por el terreno y le aparecerá el objeto, en la posición del ratón. Debajo del ratón también aparece un recuadro verde o rojo, indicando si puede o no insertar el objeto en esa posición (*fig:23b* y *fig:24a*). El objeto es posible rotarlo, haciendo click derecho del ratón o para rotar a la inversa *Shift* + botón derecho del ratón.

Una vez se ha encontrado una posición adecuada para insertar el reto, se presiona el botón izquierdo del ratón y el reto quedará fijado.

Este proceso es idéntico para todos los casos de inserción de objetos en el mapa.

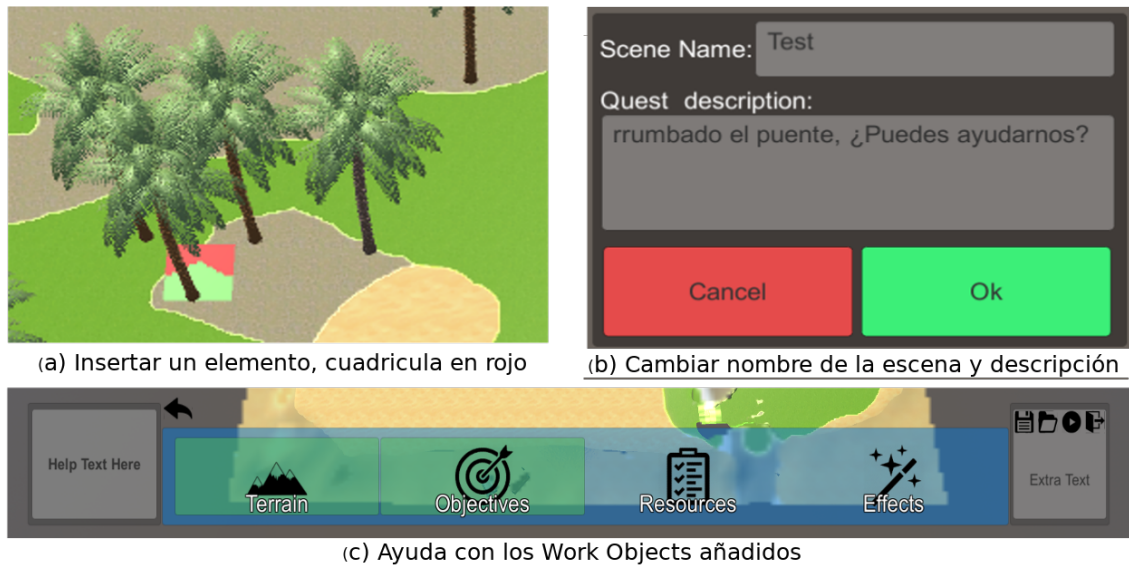


Figura 24: Insertar un elemento, modificar el nombre de la escena y progreso de los Work Objects

Como se observa en la figura 24a, al insertar cualquier elemento, aparece una cuadrícula que indica si es posible o no insertar, en esa posición, un objeto. En el caso de la imagen en esa posición no se permite añadir el objeto, porque colisiona con otros dos.

A medida que se vayan insertando elementos del Work Object, la interfaz que permite insertarlos se irá modificando. La modificación pone de color verde aquel o aquellos elementos del Work Object que se hayan insertado (*fig:24c*). De esta forma se indica al jugador que estos elementos son obligatorios y puede ver fácilmente cuáles ha añadido.

Como se ha descrito en el diseño de la interfaz, con los Mockups, en la esquina superior izquierda se encuentra el nombre de la escena. Si se aprieta ese botón se abre la ventana que permite modificar el nombre de la escena y los objetivos que tendrá el reto que se está creando (*fig:24b*).

5.2.3. Probar la escena

Una vez se han ido insertando los elementos necesarios y deseados a la escena (*fig: 25a*), es posible probarla, para comprobar como se verá en Fracsland (*fig: 25b*).

Para probar la escena, el jugador debe presionar el botón de ejecución en la sección de botones de configuración.



Figura 25: Escena terminada y probando la escena

5.2.4. Guardar y cargar escena

En cualquier momento es posible guardar la escena o cargar una escena ya hecha.

Para guardar la escena, el usuario, se dirige a la sección de botones de configuración, en la interfaz principal o de Work Objects se encuentra en la parte inferior derecha y en las interfaces de selección del tipo de elemento se encuentra en centro de la parte inferior. Una vez encontrado esta sección, se presiona el botón. Cuando se presiona el botón aparece un explorador de archivos, desde este explorador se puede seleccionar donde guardar la escena, en formato JSON.

Para cargar la escena, es igual que guardar, pero en esta ocasión se indica donde se encuentra la escena a cargar, el archivo JSON.

5.3. Resultados de los test de usuario

En este apartado se explican y se analizan los resultados de los test de usuario que se han realizado.

5.3.1. Resultados del mod

El test de usuario se ha realizado a un total de 5 personas, con las siguientes edades:

Edad	36	29	23	30	23
Media de edad	28.2				

Cuadro 1: Edades de los usuarios del test

El promedio de los usuarios que han probado el juego, según el genero, ha sido:

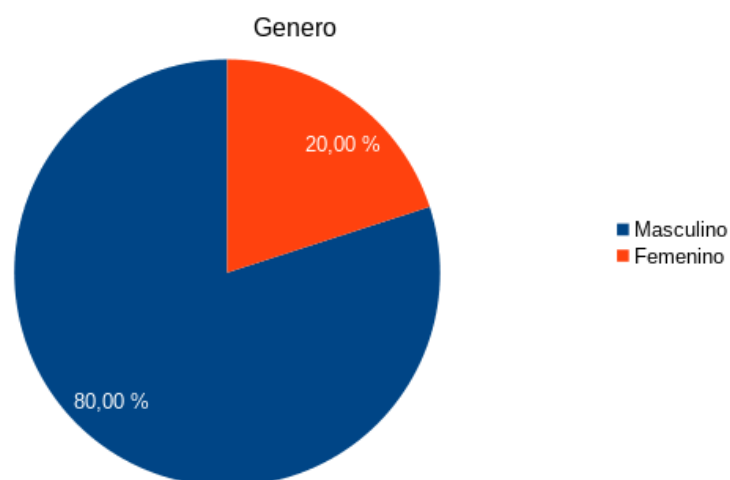


Figura 26: Genero según los usuarios del test

5.3.1.1. Preguntas de respuesta numérica

Para realizar este test de usuario, se han preguntado 5 preguntas sobre las sensaciones que han tenido (*fig: 27*):

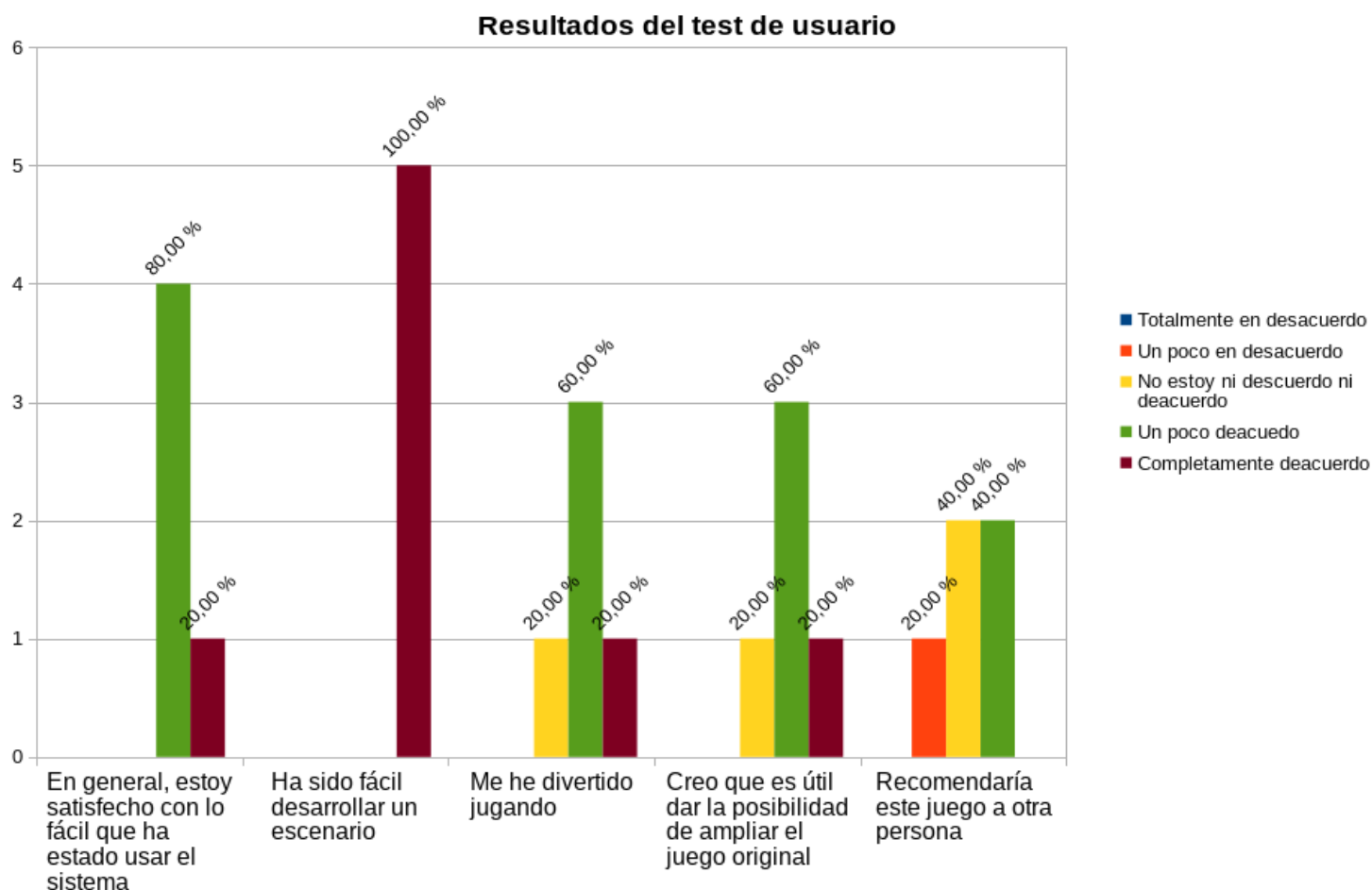


Figura 27: Resultados de los test

Como se puede ver en los resultados de los test, en general los usuarios consideran que es relativamente fácil usar el mod. Todos coinciden que desarrollar un nuevo escenario es muy fácil, así que lo que se deduce: al inicio puede ser un poco complicado de entender, pero cuando comienzan a interactuar con el juego lo entienden fácilmente y pueden desarrollar un escenario de forma rápida.

En general consideran el juego un poco divertido, pero no en exceso. Ocurre lo mismo con la pregunta de si consideran útil ampliar el juego original, en conjunto consideran que es necesario. Es decir es divertido jugar al mod, pero como es un editor de mundo no en exceso, así como ampliar el juego se considera útil pero hasta cierto punto.

Por último, en conjunto consideran que no están seguros si recomendarían el juego a una tercera persona.

Probablemente el hecho de que no consideren muy útil ampliar el juego original, así como, recomendar este mod sea debido a que no ha sido integrado con el juego original.

5.3.1.2. Resultados de las preguntas de redactar

Se han realizado dos preguntas que debían redactar para explicar que es lo que les ha gustado más y lo que menos.

Creo que este juego ha de mejorar en:
"Ha de mejorar la jugabilidad. Tiene mucha flexibilidad en la edición, pero una vez creado el mapa no se puede interactuar con el mismo."
"Movilidad del personaje dentro del juego."
"Calidad gráfica y animación en general, así como en cantidad de objetivos."
"En los menús de la derecha poner todas las figuras en color para así no crear confusión de si están o no activadas."
"Jugabilidad."

Cuadro 2: Respuestas de cosas a mejorar

Como se observa, en general, coinciden que se debe mejorar la jugabilidad para probar el juego una vez terminada la escena. También se menciona que haya más objetivos que puedan ser seleccionados. Y una aclaración de los iconos laterales.

Lo que más me ha gustado de este juego es:
"Me ha gustado la gran variedad de opciones para editar el mapa. También me ha impresionado la capacidad para editar el color de las texturas, incluso se puede escoger un color específico con la paleta de colores."
"Poder diseñar el mapa tanto a partir de objetos como en los colores de estos."
"Lo fácil que ha sido seguir los pasos del juego para poder avanzar en él."
"La simplicidad a la hora de crear un nuevo escenario."
"Posibilidad de elegir."

Cuadro 3: Respuestas de lo que más ha gustado

A la mayoría de jugadores les ha gustado la cantidad de opciones disponibles para editar el mapa, así como lo fácil que les ha resultado crear una nueva escena. Además se menciona la posibilidad de ampliar la cantidad de texturas con la selección de un color en concreto.

6. Conclusiones y trabajo futuro

Tal y como se mencionó anteriormente, esta aplicación trataba de alargar la vida de Fracsland dando la posibilidad al usuario de crear nuevos mapas.

En el transcurso del proyecto se han podido completar muchos de los objetivos planteados inicialmente.

Objetivos conseguidos:

- Se ha creado una plataforma interactiva.
- En esta plataforma se ha conseguido insertar diferentes tipos de elementos para crear una escena. Como se ha ido mencionando, los elementos interactivos son: El terreno, la misión a completar, materiales de la misión, efectos del reto, materiales decorativos, personajes no jugables y animales, cambiar el cielo y cambiar algunas texturas.
- La posibilidad de guardar y cargar la escena.
- Crear una prueba de funcionamiento con la escena creada.

En los objetivos iniciales se planteó el objetivo que permita integrar estos nuevos escenarios al juego base de Fracsland. Este objetivo no se ha conseguido en su totalidad. En principio, el proyecto que se ha creado permite crear nuevos escenarios con sus retos, recursos necesarios, objetos decorativos, cielos de las escenas, etc. y guardarlos en ficheros JSON que podrán ser cargados como escenarios nuevos en Fracsland. Sin embargo ha quedado por ampliar el código de Fracsland para poder cargar dichos escenarios con sus retos e integrarlos en la lista de misiones ofrecidas por Lord Barus.

En todo momento se ha tenido presente que este mod se ha de poder integrar en Fracsland. Así mismo todos los mapas creados por el mod deben ser capaces de ser leídos e interpretados por Fracsland.

El problema principal que se ha encontrado es que el hecho de crear un editor que permita modificar el mundo de Fracsland es una buena idea, pero complicada de llevar a cabo. Es complicado porque no es habitual hacer un editor de mundo con Unity, esto complica las cosas, al no haber demasiada información de cómo poder desarrollar este concepto.

Sin duda es posible crear un editor de mundo, porque hay muchas empresas que usan este motor gráfico (Blizzard Entertainment, Ubisoft, etc), pero son equipos grandes con gente muy especializada. En este proyecto se ha estudiado para cada componente la forma de desarrollar cada una de las utilidades del mod, sin tener demasiada información al respecto y, en ocasiones, probar desarrollos que parecían ofrecer una solución, pero que al ser integrados no resultaban ser útiles en este contexto.

Como líneas futuras de ampliación y continuación del proyecto, se pueden considerar las siguientes:

- Se puede crear una plataforma donde subir los mapas editados por la comunidad.
- Modificar Fracslan y este proyecto para que se puedan obtener y subir los mapas que se van diseñando.
- Dar la posibilidad de crear nuevos terrenos, y que el jugador cree el terreno. Así como la posibilidad de importar elementos hechos por la comunidad
- La posibilidad de crear nuevos tipos de retos, es decir que sean sumas, restas, etc. o incluso para enseñar lenguas.

7. Apéndice A: Manual Técnico

7.1. Instalación del entorno de desarrollo: Requisitos mínimos y pasos a seguir

7.1.1. Requisitos mínimos

Para poder desarrollar el juego se necesita tener instalado *Unity 3D v5.5.0f3* o superior y *Visual Studio Community 2017* o superior o en su lugar *MonoDevelop*.

Requisitos mínimos Unity 3D .

Para desarrollo

OS: Windows 7 SP1+, 8, 10; Mac OS X 10.8+.

Windows XP y Vista no son compatibles; y las versiones de servidor de Windows & OS X no se han probado.

GPU: Tarjeta gráfica con DX9 (modelo de shader 3.0) o DX11 con capacidades de funciones de nivel 9.3.

El resto depende principalmente de la complejidad de sus proyectos.

Requisitos adicionales para el desarrollo de plataformas:

- iOS: Computadora Mac con sistema operativo versión OS X 10.9.4 como mínimo y Xcode 7.0 o superior.
- Android: Android SDK y Java Development Kit (JDK).
- Windows Store: Windows 8.1 (64 bits) y Visual Studio y plataforma SDK correspondiente:
 - Windows 8.1/Windows Phone 8.1: Visual Studio 2013 o versión posterior y Windows 8.1 SDK;
 - UWP: Visual Studio 2015 o versión posterior y Windows 10 SDK;
 - El scripting backend IL2CPP también requiere la instalación de la función de compilador C++ con Visual Studio.
- WebGL: Mac OS X 10.8+ o Windows 7 SP1+ (solo editor de 64 bits).

Figura 28: Requisitos mínimos Unity 3D

Rquisitos mínimos Visual Studio Community 2017 .

Visual Studio 2017

Los siguientes productos admiten los requisitos mínimos del sistema siguientes:

- Visual Studio Enterprise 2017
- Visual Studio Professional 2017
- Visual Studio Community 2017
- Visual Studio Test Professional 2017
- Visual Studio Test Agent 2017
- Visual Studio Test Controller 2017
- Visual Studio Team Foundation Server Office Integration 2017
- Visual Studio Feedback Client 2017

Sistemas operativos admitidos	<p>Visual Studio 2017 se instalará y ejecutará en los siguientes sistemas operativos:</p> <ul style="list-style-type: none">• Windows 10 versión 1507 o superior: Home, Professional, Education y Enterprise (LTSC no se admite)• Windows Server 2016: Standard y Datacenter• Windows 8.1 (con actualización 2919355): Basic, Professional y Enterprise• Windows Server 2012 R2 (con actualización 2919355): Essentials, Standard y Datacenter• Windows 7 SP1 (con las actualizaciones más recientes de Windows): Home, Premium, Professional, Enterprise y Ultimate
Hardware	<ul style="list-style-type: none">• Procesador de 1,8 GHz o superior. Doble núcleo o superior recomendado.• 2 GB de RAM; 4 GB de RAM recomendado (mínimo de 2,5 GB si se ejecuta en una máquina virtual)• Espacio en disco duro: entre 1 GB y 40 GB, dependiendo de las características instaladas.• Tarjeta de vídeo que admita una resolución de pantalla mínima de 720p (1280 x 720); Visual Studio funcionará mejor con una resolución de WXGA (1366 x 768) o superior.

Figura 29: Requisitos mínimos Visual Studio Community 2017

7.1.2. Instalación entorno de desarrollo del juego

Para trabajar en este juego se necesita tener instalado el *Unity 3D*, *Visual Studio* o *MonoDevelop*. **Unity 3D** para la parte gráfica del juego, y **Visual Studio** o **MonoDevelop** para la parte de programación, ya que está desarrollada en C#.

Para obtener los programas dirigirse a:

- Unity 3D: <https://store.unity.com/es/>
- Visual Studio: <https://www.visualstudio.com/es/downloads/>

Una vez obtenidos los programas, hay que proceder a instalarlos. Para evitar problemas, el orden correcto ha instalar es:

1. *Visual Studio*, siguiendo los pasos del instalador.
2. *Unity 3D*, igual que el programa anterior, siguiendo los pasos del instalador.

7.1.3. Importar el juego a la versión de Unity

Cuando se han instalado los programas se puede proceder a importar el proyecto a *Unity*.

Primero se debe clonar el proyecto desde el repositorio público donde está alojado en <https://github.com/deceiver91/WorldEditor>

Para importar el proyecto correctamente se deben seguir los siguientes pasos:

1. Abrir *Unity 3D*
2. Hacer click en *Open*, se busca y selecciona el proyecto
3. Una vez abierto *Unity*, ir a *Assets* → *Run API Updater ...*
4. Comprobar que no hay errores y si no los hay, ya se puede trabajar en el proyecto.

7.2. Instalación del juego para usuarios

Los requisitos mínimos para instalar el juego son los mismos que para el entorno de desarrollo, el apartado de Unity.

Pasos a seguir para instalar el juego:

- En primer lugar el jugador debe descargar la última versión disponible, para el sistema operativo deseado. El juego está disponible en: <https://github.com/deceiver91/WorldEditor/releases>.
- Una vez descargado, se debe descomprimir el archivo .zip.
- Dentro del archivo descomprimido, hay un ejecutable. Se hace doble click en el ejecutable y ya estará el juego en marcha.

Referencias

- [1] A. 12+. DragonBox Algebra 12+. URL <http://dragonbox.com/products/algebra-12>.
- [2] J. V. P. Alfonso. ¿Qué son los “Serious Games” (juegos serios)? - ExeBlog, desarrollo de videojuegos y más. URL <http://www.exelweiss.com/blog/356/serious-games-juegos-serios/>.
- [3] E. Alhadeff. Serious Games Merge Time Tables With Epic Adventures, . URL <https://seriousgamesmarket.blogspot.com.es/2013/05/serious-games-merge-time-tables-diving.html>.
- [4] E. Alhadeff. Serious Games As A Playground For Learning Math, . URL <https://seriousgamesmarket.blogspot.com.es/2013/12/serious-games-as-playground-for.html>.
- [5] S. Blizzard. Blizzard Entertainment: StarCraft, . URL <http://eu.blizzard.com/es-es/games/sc/>.
- [6] W. Blizzard. World of Warcraft, . URL <https://worldofwarcraft.com/en-gb/>.
- [7] D. Crawley. 12 games that teach kids to code — and are even fun, too, June 2014. URL <https://venturebeat.com/2014/06/03/12-games-that-teach-kids-to-code/>.
- [8] Duolingo. Aprende un idioma completamente gratis. URL <https://es.duolingo.com/>.
- [9] Enciclográfica. La Banda de Moebius. URL <https://www.sitographics.com/conceptos/notas/moebius.html>.
- [10] B. Entertainment. Blizzard Entertainment. URL <http://eu.blizzard.com/es-es/>.
- [11] L. Estirado. ‘Survival’, un videojuego para concienciar sobre los refugiados, Apr. 2017. URL <http://www.elperiodico.com/es/noticias/extra/survival-videojuego-refugiados-5992502>.
- [12] B. D. Game. Fractions dolphin racing game. URL <http://www.bbc.co.uk/skillswise/game/ma17frac-game-dolphin-racing-fractions>.

- [13] Gamelearn. Todo lo que necesitas saber sobre los serious games y el game-based learning, explicado con ejemplos, Mar. 2017. URL <https://game-learn.com/lo-que-necesitas-saber-serious-games-game-based-learning-ejemplos/>.
- [14] GamerDic. Mod | Definición en GamerDic | Diccionario de términos sobre videojuegos y cultura gamer. URL <http://www.gamerdic.es/termino/mod>.
- [15] M. Games. Fraction Games. URL <http://www.maths-games.org/fraction-games.html>.
- [16] Kiupe. Multiplications with Math Mathews. URL <http://kiupe.com/games/multiplications-with-math-mathews/>.
- [17] R. Klacan. Draw grid on the terrain in Unity – René Klačan. URL <http://rene.klacan.sk/unity3d/games/2015/01/23/draw-grid-on-the-terrain-in-unity/>.
- [18] math play ground. Fraction and Decimal Games | MathPlayground.com. URL https://www.mathplayground.com/index_fractions.html.
- [19] I. N. Mathbreakers. Mathbreakers. URL <https://www.mathbreakers.com/>.
- [20] Mojang. Official site. URL <https://minecraft.net/en-us/>.
- [21] C. Muriel Ordóñez. Fracsland: joc seriós per aprendre fraccions. June 2016. URL <http://diposit.ub.edu/dspace/handle/2445/104290>.
- [22] pbskids.org. Math Games for Kids . Cyberchase | PBS KIDS. URL <https://pbskids.org/cyberchase/math-games/melvins-make-match>.
- [23] pipoclub.com. Juegos Para Niños Gratis en Pipoclub | Fracciones. URL <https://www.pipoclub.com/juegos-para-ninos-gratis/juego-fracciones.html>.
- [24] M. d. C. Ruiz. Tangram | Figuras para imprimir online, Apr. 2016. URL <http://webdelmaestro.com/tangram-figuras-imprimir-online/>.
- [25] M. Sauval. Michel Sauval - Jacques Lacan - Seminario 10 - La angustia - Moebius II. URL <http://www.sauval.com/angustia/moebius2.htm>.
- [26] Scratch. Scratch - Imagine, Program, Share. URL <https://scratch.mit.edu/>.

- [27] sinfallas. ¿Cuál es el significado de MMOG y MOBA?, Apr. 2015. URL <https://sinfallas.wordpress.com/2015/04/09/cual-es-el-significado-de-mmog-y-moba/>.
- [28] A. G. Unity. Unity - Adventure Game Tutorial, . URL <https://unity3d.com/es/learn/tutorials/projects/adventure-game-tutorial>.
- [29] G. E. Unity. Unity - Game Engine, . URL <https://unity3d.com>.
- [30] U. C. Unity. Unity - UI Canvas, . URL <https://unity3d.com/es/learn/tutorials/topics/user-interface-ui/ui-canvas>.
- [31] Wikipedia. Defense of the Ancients, Apr. 2017. URL https://es.wikipedia.org/w/index.php?title=Defense_of_the_Ancients&oldid=98354439. Page Version ID: 98354439.
- [32] Wikipedia. Sudoku, Apr. 2017. URL <https://es.wikipedia.org/w/index.php?title=Sudoku&oldid=98448354>. Page Version ID: 98448354.